THE USENIX ASSOCIATION NEWSLETTER

;login:

APRIL 1995 • VOLUME 20 • NUMBER 2

CONTENTS

FROM THE EDITOR

OPINION

USENIX NEWS

SAGE NEWS

FEATURES

STANDARDS REPORTS

BOOK REVIEWS

ANNOUNCEMENTS & CALLS

ETC...

USENIX The UNIX and Advanced Computing Systems Professional and Technical Association

;login: is the official newsletter of the USENIX Association.

- *;login:* (ISSN 1044-6397) Volume 20, Number 2 (April 1995) is published bimonthly by the USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.
- $24 of each member's annual dues is for an annual subscription to *;login:*. Subscriptions for nonmembers are $50 per year.
- Second-class postage paid at Berkeley, CA and additional offices.
- POSTMASTER: Send address changes to *;login:*, USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.

EDITORIAL STAFF

- Rob Kolstad, Editor *<kolstad@usenix.org>*
- Tina Darmohray SAGE Editor *<tmd@usenix.org>*
- Nick Stoughton, Standards Report Editor *<nick@usenix.org>*
- Carolyn S. Carr, Managing Editor & Typesetter *<carolyn@usenix.org>*
- Monica Jensen, Copy Editor
- Eileen Curtis, Proofreader

MEMBERSHIP AND PUBLICATIONS

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710
Telephone: 510/528-8649
FAX: 510/548-5738
Email: *<office@usenix.org>*

CONTRIBUTIONS SOLICITED

You are encouraged to contribute articles, book reviews, and announcements to *;login:*. Send them via email to *<login@usenix.org>* or through the postal system to the Association office. Send SAGE material to *<bigmac@erg.sri.com>*. The Association reserves the right to edit submitted material. Any reproduction of this newsletter in its entirety or in part requires the permission of the Association and the author(s).

;login: is produced with Framemaker 4.0 software provided by Frame Technology and displayed on an NCD X Terminal, donated by Network Computing Devices. The system is served by a Sun SPARCsystem 10 server. Output is sent to a 600 dpi QMS 860 laser printer, donated by Quality Micro Systems. The newsletter is printed on recycled paper.

SAVE THESE DATES!

Upcoming USENIX Events

2nd USENIX Symposium on Mobile and Location-Independent Computing

April 10-11, 1995, University of Michigan-Campus Inn, Ann Arbor, MI
Program Chair: Jim Rees, CITI, University of Michigan

4th UNIX System Administration, Networking and Security Symposium (SANS IV)

April 24-29, 1995, Washington, D.C.
Sponsored by the FedUnix and SAGE, the System Administrators Guild.
Program Chair: Rob Kolstad, BSDI, Inc.

5th USENIX UNIX Security Symposium

June 5-7, 1995, Marriott Hotel, Salt Lake City, Utah
Sponsored by USENIX, in cooperation with: The Computer Emergency Response Team (CERT), IFIP WG 11.4, and UniForum.
Program Chair: Fred Avolio, Trusted Information Systems, Inc.
Camera-ready Papers Due: May 1, 1995

USENIX Conference on Object-Oriented Technologies (COOTS)

June 26-29, 1995, Monterey Conference Center, Monterey, CA
Program Chair: Vince Russo, Purdue University
Camera-ready Papers Due: May 15, 1995

Tcl/Tk Workshop '95

July 6-8, 1995, Royal York Hotel, Toronto, Ontario, Canada
Sponsored by Unisys, Inc., and USENIX
Program Chairs: Ben Bederson, University of New Mexico and Will Wilbrink, Unisys, Inc.
Papers and Demos Accepted: April 14, 1995; Camera-ready Papers Due: May 23, 1995

USENIX Workshop on Electronic Commerce

July 11-14, 1995, Sheraton New York Hotel and Towers, New York City, NY
Program Chair: Dan Geer, Open Vision Technologies.
Abstracts Due: May 8, 1995; Authors Notified: May 22, 1995; Camera-ready Papers Due: June 15, 1995

9th USENIX Systems Administration Conference (LISA '95)

September 18–22, 1995, Monterey Conference Center, Monterey, CA
Co-sponsored by USENIX and SAGE, the System Administrators Guild
Program Chairs: Tina Darmohray, Consultant, and Paul Evans, Synopsys, Inc.
Abstracts Due: May 1,1995; Authors Notified: June 5; Camera-ready Papers Due: August 1, 1995

USENIX 1996 Technical Conference

January 22-26, 1996, San Diego Marriott Hotel & Marina, San Diego, CA
Program Chair: Bob Gray, US WEST
Abstracts Due: July 18, 1995; Authors Notified: August 31, 1995; Camera-ready Papers Due: November 14, 1995

For more information about USENIX and its events, access the USENIX Resource Center on the World Wide Web. The URL is http://www.usenix.org. Or you can send email to our mailserver at: *info@usenix.org*. Your message should contain the line: *send catalog*. A catalog will be returned to you.

FROM THE EDITOR

Sustaining Performance

I don't know many people who really look forward to growing older. I know I'm not in that set! You'll never hear me say, "Oh goodie, only four more years until I'm half a century old!"

Now that I'm treading well past youth, I'm learning all those things that my parents couldn't teach me (or that I couldn't learn, at any rate).

Of course, the good things center around stability (a growing desire that I don't recall having 20 years ago), financial security (such as it is), and professional growth. The bad things include middle age spread, coming off the physical peak (I'm way off the peak. . .), and a malaise that seems to affect me occasionally.

The most interesting thing I've learned lately is that doing something once is challenging and exciting. Sustaining it for a decade or two is not just challenging: it's difficult.

I don't quite understand it, but somehow time is ever more precious. I am head judge at the local science fair. It used to be so easy to get judges, suggest improvements, and generally help Fair operations. Nowadays, it's a struggle to keep the finely honed machine in perfect tune.

Likewise, the Space Settlement Design contest at Jet Propulsion Laboratories. Keeping it interesting ten years later is a big challenge.

Sustaining performance in general (both with the hobbies above and at work) seems to be a tough row to hoe. The more I watch other people who have stretched their youthful accomplishments into a string 20 or 30 years long, the more I am in awe of their vitality and tenacity.

RK

VHLL Symposium Report – Thanks!

To: Jeff Haemer

I did not attend the VHLL (like you, I was also thrown off by the name), but after it was over and I heard what it was really about, I became very interested in hearing what went on. So I was delighted to find your report. It was wonderfully complete. And it was unbiased, free of innuendos and unsupported asides (either that or we share the same biases) – something that must have been a significant challenge at this particular symposium. Thank you, thank you, thank you.

Oh, and the humor was hilarious – both what you recorded and in your own words. I also appreciate all the personal insights you made. It really tied things together. It's wonderful to read a trip report by someone who is knowledgeable and knows how to tell a story in written form and in an objective and yet still interesting way.

Thanks very much for your VHLL Symposium Report.

Don Libes
<libes@cme.nist.gov>

The closing dates for the next two issues of *;login:* are April 12 and June 14,1995.

With this issue we begin publishing a new column, "Opinion." If there is a subject that you feel strongly about that would be of interest to our community, we will air it here. You can send your article to *<login@usenix.org>*

OPINION

PGP, Phil Zimmermann, Life, the Universe, and so on . . .

by Greg Rose
PGP *key ID: 09D3E64D 1994/11/30*
<Greg_Rose@sydney.sterling.com>

[*Author's Note: I'm preparing this article for* ;login: *in a very short time, mostly due to sickness, so I hereby state that in places I'm using (and modifying) words written by Hugh Miller of Loyola University Chicago. Hugh didn't assert copyright on the material I've used, and it is in a good cause, so I hope he doesn't mind too much.*

This article is presented for the information of members, in accordance with the Board of Directors' desire to keep the membership informed, but the opinions expressed in it are not the opinions of USENIX or its Board of Directors. Other contributions and points of view are, of course, welcome.]

It's funny that the resident Australian on the USENIX Board of Directors would write an article like this, but I sort of volunteered by being the one to bring this to their attention. Anyway, to cut a long story short, There Is Something Funny Going On, and if you aren't aware of it, we think you might want to be. If you already know about the Grand Jury indictment proceedings involving Phil Zimmermann (*prz@acm.org*) then you can stop now, but if you don't, please read on.

First, you need to know why I'm writing this. The combination of some fairly abstract mathematics, some archaic laws in the United States, and some "Pretty Good" software, has caused a situation with interesting ramifications.

Background: Public Key Cryptography

In 1976, Whitfield Diffie and Martin Hellman wrote a paper about asymmetric cryptography. If you imagine locks with keys to be analogous to conventional or symmetric cryptography, then public key or asymmetric cryptography is about a different kind of lock – where you have two keys, and one can secure the lock, but not open it, and the other can open the lock but not secure it. You keep the one that can open the lock in your pocket, but you make as many copies as you like of the other one, and give them to your friends. Now, if you want to send a secret message to your friend Alice, you can lock it away using the copy of Alice's "locking" key, and only Alice can unlock it to read it. But there is another benefit (which stretches the analogy a bit): Alice could use her key to "lock" something away, and if you can use a copy of Alice's other key to "unlock" it successfully, you know that Alice must have "locked" it. This is the essence of digital signatures.

Ron Rivest, Adi Shamir, and Len Adelman invented what is called the RSA system a few years later. It is the only currently known system in which the keys used for "locking" and "unlocking" are interchangeable, the only difference being which one you keep secret. A careful reader will have noticed that I interchanged the meanings of "locking" and "unlocking" at the end of the last paragraph.

Phil Zimmermann, in 1990, used the RSA system in a program called "Pretty Good Privacy," or PGP for short, which enables people to send secret messages to each other, or verify the authenticity of who sent it, or both. PGP is, in some sense, too good. According to the state of the art, messages encoded or signed with PGP are uncrackable and unforgeable.

(Side note: RSA is patented in the US, and some people think that Phil may have done something wrong in using a patented algorithm in PGP. However, that issue is completely irrelevant to the rest of this article, and if you think that it matters you may as well stop reading.)

This is Pretty Serious technology. It enables freedom fighters in southeast Asia to communicate securely. It may also allow terrorists in the US to communicate securely. The US has laws, under which PGP is classed as a "munition" in the same category as tanks and napalm, which prevent its export from the US. It is worth noting that, in the same sense a people in other countries can easily manufacture tanks and napalm, any decent programmer could implement RSA encryption knowing only the published algorithm. In fact, my own interest in this issue started in 1990 when I was doing exactly that in Sydney, Australia.

So what is the problem?

Somehow, PGP was illegally exported from the US, and it was almost certainly without Phil Zimmermann's knowledge. Phil is currently engaged with a US Federal Grand Jury considering his indictment.

Note that the indictment is not for actually exporting PGP himself, which the government freely admits he did not do, but for making it available in such a manner that it might get exported by someone else! The government clearly wishes to crush Phil and send a strong message about making software available on networks, especially software they don't like, even if the author takes significant care to discourage or prevent export. They wish to establish that the author is responsible for potentially illegal acts committed by others even without his knowledge or control.

This is the issue that is important and of interest to USENIX members.

Phil Zimmermann Legal Defense Fund Appeal

[*Note: This section was originally written by Hugh Miller of Loyola University Chicago, but edited by me mostly to give up-to-the minute information, so blame me for any mistakes – Greg Rose.*]

In November, 1976, Martin Hellman and Whitfield Diffie announced their discovery of public-key cryptography by beginning their paper with the sentence: "We stand today on the brink of a revolution in cryptography."

We stand today on the brink of an important battle in the revolution they unleashed. Philip Zimmermann, who encoded and released the most popular and successful program to flow from that discovery, may be about to go to court.

It has been over fourteen months now since Phil was first informed that he was the subject of a grand jury investigation being mounted by the San Jose, CA, office of US Customs into the international distribution, over the Internet, of the original version of the program. [*On January 12th, Phil's legal team met for the first time with William Keane, Assistant US Attorney for the Northern District of California, who is in charge of the grand jury investigation, in San Jose. The aim of this meeting was, I believe, to try and get the indictment proceedings stopped, but that failed, and the grinding process continues. An indictment, if one is pursued by the government after this meeting, could be handed down shortly. – Greg Rose*]

If indicted, Phil would likely be charged with violating statute 22 USC 2778 of the US Code, "Control of arms exports and imports." This is the federal statute behind the regulation known as ITAR, "International Traffic in Arms Regulations," 22 CFR 120.1 et seq. of the Code of Federal Regulations. Specifically, the indictment would allege that Phil violated 22 USC 2778 by exporting an item listed as a "munition" in 22 CFR 120.1 et seq. without having a license to do so. That item is cryptographic software – PGP.

At stake, of course, is far more than establishing whether Phil violated federal law or not. The case presents significant issues and will establish legal precedent, a fact known to everyone involved. According to his lead counsel, Phil Dubois, the US government hopes to establish the proposition that anyone having anything at all to do with an illegal export – even someone like Phil, whose only involvement was writing the program and making it available to US citizens and who has no idea who actually exported it – has committed a federal felony offense.

The government also hopes to establish the proposition that posting a "munition" on a BBS or on the Internet is exportation. If the government wins its case, the judgment will have a profound chilling effect on the US software industry, on the free flow of information on the emerging global networks, and in particular upon the grassroots movement to put effective cryptography in the hands of ordinary citizens. The US government will, in effect, resurrect Checkpoint Charlie – on the Information Superhighway.

We may not all know the price Phil has had to pay for his courage and willingness to challenge the crypto status quo. For years now Phil has been the point man in the ongoing campaign for freely available effective cryptography for the everyday computer user. The costs, personal and professional, to him have been great. He wrote the original code for PGP 1.0 by sacrificing months of valuable time from his consulting career and exhausting his savings. He continues to devote large amounts of his time testifying before Congress, speaking at engagements around the world, and agitating for "cryptography for the masses," largely at his own expense.

Phil's legal team consists of his lead counsel, Philip Dubois of Boulder, CO; Kenneth Bass of Venable, Baetjer, Howard & Civiletti, in Washington, DC, first counsel for intelligence policy for the Justice Department under President Carter; Eben Moglen, professor of law at Columbia and Harvard Universities; Curt Karnow, a former assistant US attorney and intellectual property law specialist at Landels, Ripley & Diamond in San Francisco; and Thomas Nolan, noted criminal defense attorney in Menlo Park.

While this is a stellar legal team, what makes it even more extraordinary is that several of its members have given their time for free to Phil's case. Still, while their time has been donated so far, other expenses – travel, lodging, telephone, and other costs – have fallen to Phil. If the indictment is handed down, time and costs will soar, and the members of the team currently working pro bono may no longer be able to. Justice does not come cheap in this country, but Phil deserves the best justice money can buy him.

This is where you and I come in. Phil Dubois estimates that the costs of the case will run from US$100,000 - $150,000 [*if an indictment is handed down*], leaving aside the lawyers' fees. If Phil's team must charge for their services, the total cost of the litigation may range as high as $300,000. The legal defense fund is already several thousand dollars in the red.

Phil has assumed the burden and risk of being the first to develop truly effective tools with which we all might secure our communications against prying eyes, in a political environment increasingly hostile to such an idea – an environment in which Clipper chips and digital telephony bills are our own government's answer to our concerns. Now is the time for us all to step forward and help shoulder that burden with him.

It is time more than ever. I call on all of us, both here in the US and abroad, to help defend Phil and perhaps establish a groundbreaking legal precedent. PGP now has an installed base of hundreds of thousands of users. PGP works. It must – no other "crypto" package, of the hundreds available on the Internet and BBS's worldwide, has ever been subjected to the governmental attention PGP has. How much is PGP worth to you? How much is the complete security of your thoughts, writings, ideas, communications, your life's work, worth to you? The price of a retail application package? Send it. More? Send it. Whatever you can spare: send it.

A legal trust fund, the Philip Zimmermann Defense Fund (PZDF), has been established with Phil Dubois in Boulder. Donations will be accepted in any reliable form, check, money order, or wire transfer, and in any currency, as well as by credit card.

You may give anonymously or not, but please – give generously. If you admire PGP, what it was intended to do and the ideals which animated its creation, express your support with a contribution to this fund.

How to donate

To send a check or money order by mail, make it payable to "Philip L. Dubois, Attorney Trust Account." Mail the check or money order to the following address:

Philip Dubois
2305 Broadway
Boulder, CO USA 80304
(Phone #: +1-303-444-3885)

To send a wire transfer, your bank will need the following information:

Bank: VectraBank
Routing #: 107004365
Account #: 0113830
Account Name: "Philip L. Dubois, Attorney Trust Account"

Now here's the neat bit. You can make a donation to the PZDF by Internet mail on your VISA or MasterCard. Worried about snoopers intercepting your e-mail? Don't worry: use PGP.

Simply compose a message in plain ASCII text giving the following: the recipient ("Philip L. Dubois, Attorney Trust Account"); the bank name of your VISA or MasterCard; the name which appears on it; a phone number at which you can be reached in case of problems; the card number; date of expiry; and, most important, the amount you wish to donate. (Make this last item as large as possible.) Then use PGP to encrypt and ASCII-armor the message using Phil Dubois's public key, enclosed below. (You can also sign the message if you like.) Email the output file to Phil Dubois (d*ubois@csn.org*). Please be sure to use a "Subject:" line reading something like "Phil Zimmermann Defense Fund" so he'll know to decrypt it right away. You can easily find out how to get PGP and Phil Dubois' public key if you want to, just see the various FAQs in *sci.crypt* and *alt.security.pgp*.

[I don't know of any opposing points of view; but we'd be delighted to print them as appropriate . . . RK]

What is USENIX going to do?

The USENIX Board of Directors thought fairly hard about whether or not the Association could financially support Phil's defense. It turns out that we cannot, regardless of whether we wanted to or not, due to the strong possibility that doing so would jeopardize our tax-exempt status as a 501 (c) 3 organization, whose primary purpose is education. It is not clear that a donation of this type falls within the USENIX charter, and the high costs for doing the necessary legal research to investigate other like-cases would be prohibitive. However, keeping our members advised about the status of this issue is certainly educational. And to this we end, we intend to:

- Write a regular piece in this newsletter; keep you advised of the state of affairs. (Generally the articles won't be this long, but the first one needed the background.)
- Get "snitch reports" of the indictment and, if it goes that far, prosecution, along the lines of the ones we do for the POSIX committees, and publish these in the newsletter and as press releases.
- Prepare "Friend of the Court" submissions on the subject if that appears relevant.
- Possibly release a position paper to the press.
- Let you know how to donate yourself, if that is something you want to do. I have.

PPG Offers Free Email Newsletters

Parallel Performance Group (PPG) would like to announce a Series of FREE monthly Email Newsletters on high-tech software topics.

Below is a partial list of the topics discussed in the series of newsletters:

Volume Visualization
Real-Time Multiprocessing/Uniprocessing DSP
Multiprocessing
Neural Networks/Fuzzy Logic/Fuzzy Control
Parallel Processing
Animation/Solid Modelling
Distributed Processing
Mechanical Design Automation
Object-Oriented CASE Tools
CAD for Industrial and Architectural Design
Project Management
Discrete and Continuous Simulation
Image Processing
Medical Imaging

To receive more information, please send any email to *info@ppginc.com*; this will return a more detailed listing of the Newsletters, and instructions on how to receive them. You will not be put on any mailing list unless you specifically ask to be.

USENIX MEMBER BENEFITS

As a member of the USENIX Association, you receive the following benefits:

- Free subscription to *;login:*, technical features, system administration tips and techniques, international calendar of events, SAGE News, media reviews, Snitch Reports from the USENIX representative and others on various ANSI, IEEE, and ISO standards efforts, and much more

- Free subscription to *Computing Systems*, the refereed technical quarterly published with The MIT Press

- Discounts on registration fees for the large, multi-topic technical conference, the System Administration conference (LISA), and the various single-topic symposia addressing topics such as object-oriented technologies, security, operating systems, electronic commerce, and mobile computing – as many as seven technical meetings every year

- Discounts on proceedings from USENIX conferences and symposia and other technical publications. Discounts on the USENIX Association book published by The MIT Press. Now available: *The Evolution of C++: Language Design in the Marketplace of Ideas*, edited by Jim Waldo of Sun Microsystems Laboratories

- Discounts on BSDI, Inc. products. BSDI information: 800 800-4BSD

- Discounts on the five volume set of 4.4 BSD manuals plus CD published by O'Reilly & Associates and USENIX

- Savings (10-20%) on selected publications from McGraw-Hill, The MIT Press, Prentice Hall, John Wiley & Sons, O'Reilly and Associates, and UniForum

- Special subscription rates to *The LINUX Journal*. Phone: 206 527-3385

- Right to join SAGE, the System Administrators Guild

To become a member or receive information regarding your membership status or benefits, please contact <*office@usenix.org*>. Telephone 510 528-8649.

USENIX NEWS

1995 USENIX Technical Conference Reports

New Orleans, Louisiana

(*Note: The following reports cover most but not all of the refereed papers and invited talks. Thanks to the reviewers: Win Bent, Peter Collinson, Bryan Costales, Rasit Eskigioglu, Kenneth Ingham, George Leach, Jerry Peek, Peter Salus, Peg Schafer, and Rick Thomas! If you wish to refer to the full papers, go to the Online Library on our Web site <URL: http://www.usenix.org> or order the conference proceedings by contacting* office@usenix.org. *If you would like to write a summary of a conference, contact* login@usenix.org.)

Congratulations to the winners! Best Paper: *Performance Implementations of Multiple Pointer Sizes* by Jefffrey C. Mogul, Joel F. Bartlett, Robert N. Mayo and Amitabh Srivastava. Best Student Paper: *File System Logging versus Clustering: A Performance Comparison* by Margot Seltzer, Keith A. Smith, Hari Balakrishnan, Jacqueline Chang, Sara McMains and Venkata Padmanabhan.

Keynote Address

Summarized by Peter H. Salus
<phs@netcom.com>

Mark Weiser of Xerox PARC delivered the keynote address in New Orleans. Entitled "Ubiquitous Computing," I found it largely unimaginative and pretentious. Weiser can be an entertaining speaker, but his topic and lack of vision restrained him on January 18.

Weiser began by telling his audience that ubiquitous computing had "something to offend everyone" and proceeded to link UC to the history of western civilization, calling it the "third wave" in computing: first, many people working on a single computer; then, one system per person; next, "hundreds of machines" per person. He then made a quip ("The OS of the fifties will be out next year from Microsoft") and delivered himself of the unremarkable "Good technology is invisible."

Weiser next turned to fitting his concept of UC into Western Civ by misconstruing both Michael Polanyi (proximal vs. digital) and Stephen Toulmin (modernism vs. postmodernism) and latching on to the slogan of "empowerment." He also cited the "feminist/humanist" approach of deconstruction.

Having touched what he saw as the appropriate buttons, Weiser broached his vision of three sizes of devices (tabs, pads, and boards) which communicated via infrared. He showed us a videotape and discussed the notion of measuring computing power in terms of MIPJ (per Joule) rather than MIPS.

Weiser went on to tell the thousand of us about smart phones that would follow our movements and smart ID cards that would track employees. I envisaged employers tracking time in the can and visits to the coffee/soda machine. I mused about the infrared fixtures in cars, buses, and subways. I wondered what would happen when I went out of doors with my pad.

In one sense computing is ubiquitous: it all depends on what you count as computing. The chips in the microwave, the dishwasher, the vcr, the cd player, the alarm clock, the car are all computers. I already have dozens at home (not counting the box I'm writing this on.)

I didn't find Weiser illuminating. Anyway, I don't want folks to track me by computer or phone most of the time. And I sure don't want infrared detectors in every room at home.

[Editor's Note: We are sure there are others who heard Mark's talk who had different impressions. We welcome other viewpoints. We have the audiotape of the address if you need refreshing. Please send your contributions/summaries to login@usenix.org.]

USENIX 20th Anniversary

In recognition of USENIX's 20 year anniversary, Steve Johnson, President and longtime USENIX luminary, gave a short outline of some of the highlights of the Association. Some of the interesting facts he shared are:

- The name USENIX was created by Margaret Law in 1978. *[Margaret was/is at Radcliffe and was/is Lew Law's spouse. Lew was on the first USENIX Board and remained on the Board till 1986 – Ed. Note.*] Contrary to popular belief, it was not a trademark of Bell Labs; in fact, USENIX avoided AT&T discomfort with the earlier *UNIX News.*
- Having Conference Proceedings available at the conference in 1985 was revolutionary. At that time, most technical conferences either produced the proceedings six months after the conference, or required the papers six to eight months before the conference.
- UUNET was spun off of USENIX to attempt to preserve USENET, which was about to collapse due to the astronomical phone bills of a few backbone sites.
- USENIX has presented workshops, symposia, and seminal papers on hot topics throughout its twenty years. Notable topics included: NFS, Window systems, Micro kernel, Filesystems, Mobile Computing, Supercomputing, C++, Security, and, of course, Systems Administration.
- SAGE, the first professional group for system administrators, was formed in 1992.
- USENIX is in financially strong shape, with membership growing 10% annually. The focus for the future will be on advanced computing systems and techniques, including, but not limited to, UNIX.

BSD

Summarized by Peter Collinson <pc@hillside.co.uk>

Portals in 4.4BSD

by W. Richard Stevens, Consultant, and Jan-Simon Pendry, Sequent UK

The first technical session of the conference comprised three papers on developments in the kernel. This paper was the first of a pair coauthored by Jan-Simon Pendry. Actually, he had been cunning. He is disinclined to spoil his USENIX conferences by standing up in front of all those people in a quivering state of panic, so he managed to find coauthors who find this less worrying. Richard Stevens wrote 90% of this paper, with Jan-Simon providing input on how things worked and what measurements might be interesting. The paper is a good description of how things work, how portals can be used and how well portals perform.

Richard also did most of the talking. Jan-Simon turned the slides on the overhead projector and answered questions at the end. Incidentally, Jan-Simon and Richard only met on the morning of the talk. This is another testament to how small the Internet has made the world these days.

Portals are a way of mounting a process at some point in the filesystem. The process, a portal server, is passed information about file opens from the kernel. When a client process opens a file using a pathname that traverses the mount point, the remainder of the path is sent to a portal server to interpret as it sees fit. Assuming that the portal daemon allows it, the client program is passed a file descriptor and continues to use standard read/write/close system calls. The client program can be naive using the standard UNIX I/O system calls to send or receive a stream of bytes. The portal process is a way of adding functionality to the file system model supported by UNIX without adding kernel code.

The example that Jan-Simon has implemented provides TCP network access to naive processes. If you mount the process at say "`/p`", then it's possible to use commands like:

```
$ cat < /p/tcp/noao.edu/daytime
  Wed Jul  6 11:26:07 1994
```

Notice that it's the shell that opens the file; we could have got the `cat` program to do it. The file name is passed into the kernel which notices that "`/p`" is a mount point. In turn, the string "`tcp/noac.edu/daytime`" is passed to the portal process for interpretation. The portal process splits this string up on the slashes and decides that it should initiate a TCP connection to noac.edu using the port for the daytime service. Once the TCP connection is opened it returns a file descriptor via the kernel to the shell.

The shell is convinced that it has opened a file and sets the standard input of the "`cat`" process to that file. The "`cat`" program will simply perform read operations on standard input until end of file is reached. This will happen when the TCP connection is closed.

Setting up a listener is also simple:

```
$ cat /p/tcplisten/0.0.0.0/service
```

The "`0.0.0.0`" here is a wild-card that says "use my IP address." You specify a precise IP address if you wish. The "service" string is a name derived from `/etc/services`, or it can be a port number.

Later, there was a question about the error codes returned to the naive processes from system calls that map to network connections. The answer was that the system calls will return will errors like: ENETDOWN, ECONNABORTED and the like.

There was also a question about "`stat`". The questioner was really asking what happens if you do:

```
$ ls -l /p/tcp
```

The answer was that this was not implemented; you get a file of zero size, with mode 666, owned by root with a zero group id. This was seen to be the biggest hole in the scheme by a distinguished member of the UNIX room at AT&T Bell Labs. Jan-Simon tells me: "There is nothing to stop you from implementing '`cat /p/tcp/.active`'. since '`.active`' is not a valid DNS name. I didn't implement anything more than open, mainly because I had other plans for a user-level fileserver which would allow you to do this 'right,' and also because the vnode interface for things like getting the contents of a directory really suck. I'd prefer to make an open directory be either a pipe or a socket (same difference). This allows you to do much more interesting things on the far (server) end."

Dynamic Vnodes – Design and Implementation

by Aju John, Digital Equipment Corp.

This was the second talk in the kernel developments session. It qualifies for the "inode of the conference" award. Every UNIX conference needs a talk on inodes to make the old-timers feel fulfilled. This was it.

The "inode" is the on-disk block of data that describes a file. When you open the file, the inode is brought into memory and, these days, becomes a "vnode" or "virtual inode." The "vnode" can also describe remote files for NFS.

The early kernels tended to use statically sized tables to store their resources. This lead to messages like "Out of inodes" and a system panic when the resources were exhausted. Recent kernels have simply grabbed more kernel memory when they run out of resources. So if there is a sudden peak demand for vnodes, then the kernel occupies more memory. However, these kernels don't release the resources when the peak demand vanishes – made hard to do by the way that in-memory vnodes are used. The work described in this talk has implemented the dynamic growth and freeing of vnode resources.

Why is this "hard?" Well, traditionally kernels have used the vnode table as a cache. When you open a file, the kernel translates the file name into a (device, inode) pair and will read the inode into the kernel making a vnode. When you have finished with the file, the vnode is marked as unused. You can save a disk read if another open for the same file comes along and the kernel finds the "old" copy of the vnode in the table.

OK. What's so hard about this? When you want to lose vnode structures you can just pick the ones that have been lying around the longest and throw them away. This is complicated because the kernel maintains a cache of names that are used for quick pathname translation. The design of this code has always assumed that the vnodes are available in the kernel. Although there are links from the name cache to the vnode table there are not any reverse pointers. In fact, an attempt was made to add reverse pointers, but this failed because several name cache entries can point at the same vnode.

So, if you decide to invalidate a vnode, you cannot simply look at the vnode table and throw things away. This would mean that you would have to search the name cache table for references, and it's not economic to do that. The key is to tie vnode deallocation to the name cache lookup code.

A time stamp is introduced into the name cache, and the time on each entry is compared against a kernel constant during the normal name cache searching operation. If the name cache entry has timed out, then the cache lookup operation forces a `miss' and loses the names in the cache on the assumption that the vnode has been eliminated from memory. The code which cleans the vnode table ensures that any vnode will not be eliminated if it has been resident for less time than the time specified in the kernel constant.

There are several other complications with locking and races that are explained in the paper.

The author then went on to do some performance measurements on the new system. He found little difference between the new and the old systems. He found little or no performance drop when vnode deallocation was enabled.

Union Mounts in 4.4BSD-Lite

by Jan-Simon Pendry, Sequent UK,
and Marshall Kirk McKusick, Author and Consultant

The third talk in the kernels session was a paper written by Jan-Simon Pendry with Kirk McKusick providing authoring and editorial experience and also presenting the work. Jan-Simon got some more practice at the overhead projector; his ambition is to become a professional slide turner.

Union mounts are another application for the new file system switch code implemented in 4.4BSD. This supports the notion of stackable file systems. The basic idea of the Union File system is simple. You mount one file system "on top" of another. The file systems are laid on top of each other transparently. When you traverse the structure, you can see files and directories from both file systems. You can have several layers, forming a composite file system. [*This shares some aspects of Sun's translucent filesystem . . . Ed.*]

File systems that are mounted "underneath" are read-only. All new files are created only in the top layer. When you modify a file that exists in a lower layer it is automatically "copied up" to the top layer so that the contents can be changed.

The namespaces of all the file systems in the layers are unified with duplicate names being suppressed. A name lookup for a file will locate the topmost object with the desired name. The lookup moves down a logical stack of mounted directories (the ordering can be different from the physical mount order) looking for names that match.

There are a couple of special cases: ".." is made possible by adding a new field to the vnode which points up the tree, and directories are treated specially. When a directory is looked up in a lower layer and there is no correspondingly named directory in the upper layer, then the union file system creates one called the shadow directory. This has the effect of populating the top level with directories. The directory is created in case a copy-up operation is needed. Jan-Simon tells me that these are "implementation details" and will change in future releases of the code.

Then we have the case of file deletion. The lower layers are read-only; if a file is deleted, there must be some way to give the appearance that a file has been removed from a directory. This is done by "whiteouts." A whiteout is implemented by placing a directory entry with a matching name, and tagging the name with a flag. In the new filesystem implementation, directory entries are tagged anyway. No inode is created for the entry; it's just a marker in the file system tree.

OK, so what use is all this? Well, it enables you to overlay a hard disk on top of a CDROM ISO-9660 file system and type "make" with the ".o" files appearing in the same directory rather than having to mess with makefiles that point out of the directory. It also enables you to compile the same source for different architectures by placing a number of union mounted subtrees over the source directory. It enables you to have private copies of a source directory where the lowest level is the unchanged original and top levels are your copy. It's the CDROM application that I am interested in.

This paper described recent work. The code exists in an early form in the existing 4.4BSD-Lite distribution. The paper describes an implementation which will be made available on a forthcoming 4.4BSD-Lite II release.

Mass Storage

Summarized by Bryan Costales
<bcx@lady.Berkeley.edu>

Evaluation of Design Alternatives for a Cluster File System

by Murthy Devarakonda, Ajay Mohindra, Jill Simoneaux, William H. Tetzlaff, IBM T. J. Watson Research Center

Murthy Devarakonda presented this paper. Based on homogeneous clusters of AIX machines, the study contrasted shared disk versus shared filesystem approaches to client/server NFS for speed and consistency. Specifically, PJFS was compared to Calypso and Calypso found superior. (Calypso stands for the Calypso Project at the IBM Thomas J. Watson Research Center).

Murthy failed to describe his underlying assumptions before leaping directly into details. Perhaps he expected the audience to have first read the paper. Once underway however, his presentation was quite clear.

During the question period he revealed that they have been working on fault tolerance, but that work is not a part of the current paper.

Multi-resident AFS: An Adventure in Mass Storage

by Jonathan S. Goldick, Kathy Benninger, Christopher Kirby, Christopher Maher, and Bill Zumach, Pittsburgh Supercomputing Center

Jonathan S. Goldick presented this paper about accessing files directly off tape (albeit slowly) with AFS. This system, called multi-resident AFS, was developed because, "tape is cheap." A file can exist in multiple places, including hard disk for fast access. Least often accessed files can be moved to slower, less expensive storage, like tape or optical disk. A generic device driver can be readily interfaced to any device. The underlying hardware does not need UNIX se-mantics. Data migration is supported but not implemented.

"We typically have been adding tens of disks, 50 gigabytes, per year." Multi-resident AFS lessens this need, and does so with a bonus. As Jonathan concluded, "we've given our users essentially infinite quota."

RAMA: Easy Access to a High-Bandwidth Massively Parallel File System

by Ethan L. Miller, University of Maryland, Baltimore Co., and Randy H. Katz, University of California, Berkeley

Ethan Miller described a new method for increasing the effective I/O speed of massively parallel processor machines. Ordinarily, a subset of the processors are equipped with disk I/O. With RAMA, each processor has a disk and each processor is responsible for an unusual kind of striping. A simple "pseudorandom," hashing algorithm determines which disk gets which block of a file.

RAMA was developed because processor speed is increasing rapidly, while disk transfer rates are increasing very slowly. "RAMA [was] designed to take advantage of storage, rather than to grudgingly support it." The use of pseudorandom block assignments has proven beneficial in balancing the I/O load across many processors.

Cryptography in the Electronic World

by Bruce Schneier, Counterpane Systems
Summarized by Peter Collinson
<pc@hillside.co.uk>

Bruce is the author of the excellent *Applied Cryptography* and presented an invited talk giving an overview of the why, what, and how of cryptography.

There are many levels of security. It ranges from security that will stop your kid sister from obtaining your secrets to security that will stop major governments from violating your privacy. His talk was about the latter.

He stressed that there is no security in obscurity. Modern cryptographic techniques are intentionally public and are tested by experts. Everyone in the field attempts to break everyone else's algorithms. If this cannot be done, then you can be fairly sure of the algorithm. Of course, you need to worry about the people or organizations who will not tell you what algorithms they have broken.

Conventional cryptography is good for encrypting files, backup disks, data links, electronic mail, fax transmissions, digital voice, digital video etc. In conventional cryptography, the sender and the receiver share a key. The sender uses the key to encrypt the message. The receiver uses the identical key to decrypt the message. Any eavesdropper does not know the key and so cannot decrypt the message. The problems here are twofold, first there is the problem of disseminating keys. Second, if you lose the key, you lose the data.

There are many algorithms that are discussed in the academic literature. Only a few of them are considered safe enough to trust by experts. Bruce favored DES, RC2, RC4, IDEA, GOST, and SKIPJACK. The best way to choose an algorithm is to pick a published one, because such an algorithm will have been scrutinized by many cryptographers.

Bruce then went on to discuss public key cryptography. For this, a different key is used for encryption and decryption. The first key is a sequence of bits that is public; everyone knows it. The second key is also a sequence of bits, it's private; only the recipient knows it. It is computationally infeasible to derive the private from the public key. "Computationally infeasible" is a term favored by cryptographers, it does not mean that this is impossible, but that it would take too long to do so.

Public-key and conventional cryptography do different jobs and are often used together. The problem is that public-key encryption is slow and cumbersome. The solution is to encrypt the message using a conventional algorithm and send the key with the message encrypted with public-key cryptography.

Public-key cryptography can also be used to verify that a message has come from a particular individual. If a message is encrypted with a private key, then only that person could have encrypted the message so this is the equivalent of signing. Anyone can decrypt the message with the public key, so anyone can verify the signature. In real-world implementations, the message is passed through a one-way hash function to generate a number which is then encrypted with the private key. This is a "digital signature."

One-way hash functions provide a fingerprint of a digital file. The idea is that it's fast to compute the hash value, but it's computationally infeasible (that term again) to generate the file from the hash value. It's also computationally infeasible to find two files that hash to the same value. Popular hash functions are: SNEFRU, MD2, MD4, MD5 and SHA. A typical hash length is 128 bits.

Bruce then went onto discuss more features of public key encryption. For more details, get hold of a copy of his book: *Applied Cryptography*, John Wiley & Sons, Inc., ISBN 0-471-59756-2.

Bruce ended with a few comments about the security "industry." There are many security programs that are available, most of them are garbage. It's like trusting snake oil. Many of these programs have "back doors" programmed in. Many implementations of DES do not properly implement the algorithm. Even proper implementations of DES usually use the weakest mode of operation. Most password protection algorithms and screen blankers can easily be bypassed. Most commercial software comes with encryption functions that can easily be cracked.

Most users don't want real security. They want something that is easy to use and will keep the idle curious from seeing their files. They want to be able to recover their data should they lose the key. If you can recover a forgotten key, then an adversary can read your encrypted files.

If you are really concerned about security, then you should know what you are doing. You should not rely on unsubstantiated claims for a product's security. As a general rule, if a company won't make their algorithm public, then the algorithm probably isn't very good.

If you want professional security, hire a cryptographer.

Potpourri I

Summarized by Kenneth Ingham <ingham@i-pi.co>

Implementing Real Time Packet Forwarding Policies Using Streams

by Ian Wakeman, Atanu Ghosh, Jon Crowcroft, University College, London; Van Jacobson, Sally Floyd, Lawrence Berkeley Laboratory

As the Internet grows so is the need for real-time packet delivery for traffic such as compressed audio and video. Ian and his coriander have shown a proof by existence that IP can handle real time – their implementation of Sally Floyd and Van Jacobson's class-based queueing is being used for part of the traffic passing over the UK-US FATpipe. Their work can also be used to counter the ATM folks who claim that 53 byte packets are necessary to do real time. I find it interesting that whenever someone claims that IP is incapable of X, someone else provides the proof that IP can handle X after all.

Scaling the Web of Trust: Combining Kerberos and PGP to provide Large Scale Authentication

by Jeffrey I. Schiller and Derek Atkins, MIT

PGP has an idea of a web of trust where you can have others vouch for the integrity of someone's public key (introducers). The problem with this web is that it does not scale well. Kerberos does not implement any public key encryption, so it cannot create digital signatures itself even though it can vouch for the identity of a person.

Given a Kerberos server, Derek Atkins and Jeffrey I. Schiller have created a key broker which will sign PGP keys and can be as trusted as the server. Now, when people wish to exchange email, they can use the Kerberos key signer as their introducer. If you keep secure backups, recovery from a compromised Kerberos server is not difficult. MIT is now using this technology for tasks such as using PGP to verify purchase orders sent via email.

Flexible and Safe Resolution of File Conflicts

by Puneet Kumar, M. Satyanarayanan, Carnegie Mellon University

The Coda File System uses optimistic replication to provide high availability in a distributed UNIX filesystem. The problem comes when the network becomes partitioned and different copies of the files are updated concurrently. Coda provides for the user to provide a resolver for each application that will deal with these conflicts.

Application-specific resolvers (ASRs) are specified in a file with a syntax similar to a Makefile. Puneet gave examples showing resolving TeX .dvi files, files used by a group-scheduling calendar utility and files created via make. Resolution of problems is not quick, taking half a second without counting the time spent in the resolver itself. He claimed that this time is not unacceptable, since normally the resolver takes longer to run, and the penalty is offset by the benefits of the flexibility.

When asked by a user how many different applications they had resolvers for, Puneet answered three – he is interested in the systems side of this, not interested in creating a collection of resolvers. I don't blame him, as the work he is doing is much more interesting than writing resolvers.

Software Methods for System Address Tracing

by J. Bradley Chen, Harvard University
Summarized by Peter Collinson <pc@hillside.co.uk>

I think that one of the most notable things to have happened in Computer Science in the last few years is the adoption of the "scientific" approach to Computing. It's no longer acceptable to write a paper that says "we did this and it worked." Papers now tend to contain a thesis, describe an implementation, then go onto measure the implementation against some criteria before generating a conclusion.

Brad Chen discussed his experience of the application of software instrumentation tools to operating system kernels.

The tools provide a detailed view of operating system activity, revealing not only how long it takes do to a particular operation, but also exactly what happens in the system and when.

The tools use software techniques to provide monitoring and tracing of the activity in the kernel and the user-level processes that the kernel runs. Although software instrumentation is tricky, the inherent flexibility and mobility of software techniques make it an important alternative for the measurement of future systems. Prior methods such as hardware monitors are often limited and can be difficult to apply.

Chen's main technique is to rewrite the executable image of the program to be traced. Logging and monitoring calls are inserted at instrumentation points, such as the beginning of basic blocks of the program or before load or store instructions. He described a system used on the DECstation 5000/200 workstation which saved the address trace in a large buffer in memory. The buffer was emptied periodically by an analysis task. He also described a more recent system for Alpha AXP systems from Digital, in which measurement events are processed immediately with no buffering at all.

Rewriting the executable image can be applied to user programs as well as the operating system kernel. After all, the binaries are generated by the same compiler. However, there is some sensitive code in kernel that cannot be rewritten mechanically. Activity for this code can be measured by inserting instrumentation by hand. The hand-instrumentation is straightforward, and only a small number of routines require the special treatment.

The next trick (and here's the science) was to ask "does the results from the measurements accurately reflect system activity?" Or in this case "Have I beaten the Heisenberg principle?" Chen discussed the techniques he used to validate measurements. The main idea was to perform a comparison between a real system and a simulator.

Chen, along with Brian Bershad of the University of Washington, applied the tracing tools to compare two versions of UNIX on the DECstation 5000/200: Ultrix, the UNIX product from Digital Equipment Corporation, and Mach 3.0, the microkernel system from Carnegie Mellon University. They found that for a given workload the Mach microkernel system executes more system instructions, and each instruction has a larger memory system penalty. These effects are due to the microkernel structure of Mach.

They also found that some design policy issues for the kernel have a large performance impact. In particular, the different page mapping policy implemented in the kernels of the two systems caused a great difference between both the user and system memory behavior. Also, Ultrix tends to spend more time in the idle loop due to the use of synchronous writes for file system meta-data.

In the second part of the talk, Chen discussed some recent work in instrumenting the OSF/1 operating system for Alpha AXP workstations using a tool called Atom. Atom is more robust than other tools; and more importantly, it has a tool-building interface that makes it much more flexible and easy to use.

Chen explained some details about what was required to adapt Atom for use with the kernel. He had also created some applications with Atom, writing some simple measurement tools for the OSF/1 system.

He classifies the tools into two types. Some tools measure system activity only. These have the advantage of being easier to write. As examples, he showed results from a kernel profiling tool, and a similar tool that integrated a cache simulation to give a profile of kernel cache miss activity.

The second type of tool integrates both user and system activity. As an example, Chen showed results from a tool that simulates a direct mapped cache with both user and system references. With such a tool, you can measure not only miss rates, but also interaction between user and system activity in the cache. Measurements with an 8 K byte cache showed that for small caches the performance impact of this interaction is not significant.

Objects

Summarized by Rasit Eskicioglu
<rasit@cs.uno.edu>

OODCE: A C++ Framework for OSF Distributed Computing Environment

by John Dilley, Hewlett-Packard Laboratories

John presented a method for developing object-oriented distributed applications using the C++ and Distributed Computing Environment (DCE) technologies. This work was motivated by experiences from early DCE prototypes. The main goals of the object-oriented DCE (OODCE) framework are: to simplify DCE application development, to enable development of DCE applications in C++ by encapsulating existing DCE functionality, and to provide support for distributed objects over DCE.

The Open Software Foundation's DCE is a distributed application development environment consisting of a programming environment and a set of services which includes communication, POSIX threads, naming, security, file system, and time. These services can be accessed through an application programming interface (API) or through administrative and user commands.

Use of the DCE provides application developers with many benefits compared with network programming using low-level primitives. These benefits include a powerful programming model, simple network programming, location independent integration of client applications, a formal interface between the client and the server, and an implicit object model. Along with its benefits, DCE presents a number of challenges, such as, a difficult and complex low-level interface. Furthermore, DCE applications contain many redundant code segments.

The OODCE framework defines a C++ class library which is used to hide the DCE complexity from application developers and a compiler which is built to convert DCE interface definitions into corresponding C++ client and server classes. The class library provides the default behavior of DCE functions as well as the necessary flexibility for customization of different behaviors.

Several areas of future work were identified as (1) enhancing the object factory interface, (2) integration of network management, and (3) exploring system/application management. The talk concluded by a brief overview of related (mainly commercial) work.

Mach-US: UNIX On Generic OS Object Servers

by J. Mark Stevenson, Carnegie Mellon University and Daniel P. Julin, ISIS Distributed Systems

Mark's talk was on the examination of the Mach-US operating system, its unique architecture, and the lessons learned through its implementation. Mach-US is a 4.3BSD UNIX binary-compatible, object oriented, and symmetric multi-server operating system which runs on top of the Mach 3.0 micro-kernel.

The rationale for this project was to examine various OS design issues, such as the effect of multiple servers on a micro-kernel, application programmer interface (API) neutral operating system services, remote method invocation (RMI) for OS system services and/or objects, client side OS computation, and UNIX API re-implementation. The talk presented the Mach-US architecture, showed its performance and current status, and discussed its unique OS features.

Mach-US achieves its flexibility by a series of object-oriented generic OS interfaces to its modular servers. Service interfaces are defined in C++ as abstract classes. Each service interface also defines the information to be transferred to the clients, as well as how and when this information should be transferred. Naming, access control, I/O, Net, asynchronous notification, and process management are among the services supplied by Mach-US.

The development of Mach-US has shown that multiple OS servers enhanced system configuration flexibility. Especially, strong subsystem separation required almost no inter-server communication, an issue which was otherwise believed to be the major drawback of the serverized operating systems. Also, redesigning the OS interface proved to be a valuable exercise. Just as it is for any large software project, object oriented technology was found to be helpful for OS implementation. Intelligent emulation libraries supplied significant flexibility and performed significant OS functions.

Finally, the talk concluded that the use of multiple OS servers on a micro-kernel is a viable approach to designing new operating systems. Decoupling API, OS service, and process boundaries gives important flexibility for research in OS interface design. API re-implementation is painful, but necessary.

Events in an RPC Based Distributed System

by Jim Waldo, Sun Microsystems Laboratories, Inc.

Jim talked about his group's work on building a distributed system that enables objects to register interest in and receive notifications of events in other objects. The work centers around merging two distinct communication techniques, Remote Procedure Call (RPC) and event notification.

RPC has been the most common technique for the construction of distributed systems. Communication using RPC is based on distributed function calls, in which control is passed from one process to another. In RPC, a call to a remote object transfers control to that object by packing the request into a message and sending it across the network. The calling object is then blocked until it receives the result from the remote object. This model follows a synchronous communication approach. The computation model of RPC is based on objects offering services to other objects, or objects containing other objects. Examples of RPC based systems include Clouds, DCE, and CORBA.

Event notification, however, is a less common model for distributed systems, in which notifications of changes to an object's state are communicated among objects. Unlike the RPC model, the event notifications are usually asynchronous in nature. In event notification based systems, the objects react to occurrences of other objects in their own way, allowing introduction of new objects to the system dynamically, without changing existing objects. Such systems were pioneered by Isis.

The talk showed how these rather orthogonal, yet simple, approaches can be combined. The major goals required to introduce the functionality of an event notification system into an RPC based distributed object system are:

basic services should be cheap both in terms of efficiency and implementation effort,

- complex features should be built on a simple base,

- levels of service should be transparent where possible, and

- events should not be introduced as an alternative to RPC, but should only be used to allow functionality that does not fit into the RPC paradigm.

Jim later explained in greater detail how these goals were met. The system is built around three basic concepts: identification of kinds of events, registration of interest in such events in some objects, and notification of the occurrences of an event.

Event types are identified by fixed-length values exported by individual objects. They are returned by calls that are part of an object's external interface. The same event type may be exported by different objects to represent different event types. Also, the same event type may be identified by different objects with different event type identifiers. Only those identifiers returned by calls to the same operation refer to the same event type.

Objects export the Event Generator interface in order to allow other objects to register interest in event classes that they export. This interface includes register and unregister operations. To register an interest, the caller supplies three pieces of information: a reference to the object that will receive any notifications of events that are part of this event class, the identifier of the event class of interest, and an identifier for the object that is interested in the event class.

The system, which is built on top of a pair of simple interfaces, is implemented in Modula-3. OMG CORBA IDL is used to describe the interfaces in the system. The implementation of the Event Generator and Event Catcher interfaces required 1,078 lines of Modula-3 code. The tests which make use of the above interfaces showed that both the same address space and separate address space event registrations and notifications were very efficient.

Jim concluded that events and RPC can coexist as long as they are used appropriately and that complex services can be built on top of simple ones.

Tcl for Internet Agents

by John Ousterhout, Sun Microsystems, Inc.
Summarized by Bryan Costales
<costales@ICSI.Berkeley.edu>

John represented the Internet as an inflexible, heterogeneous collection of machines and software. Currently the Internet has machines attached to it ranging from MACs to UNIX boxes to VMS based hosts. Clearly no high-level language now exists that would allow an application to be written such that it could run on all Internet platforms without porting.

John suggested the use of TCL (for Tool Command Language) and the associated TK toolkit satisfy that need. Applications written as TCL scripts have the advantage of running on a myriad of machines. TCL (and the SafeTCL security extensions) can be used to create a "universal" scripting language.

Several examples were presented.

Potpourri II

Summarized by Rick Thomas
<rbthomas@rutgers.edu>

Turning the AIX Operating System into an MP-capable OS

by Jacques Talbot, Bull

This paper reported on the development of AIX version 4 by two groups, one at IBM in Austin, Texas, and one at Bull in Grenoble, France, collaborating via a high-speed transatlantic link that enabled both groups to work from a single common source tree. The fact that the project succeeded is nothing short of amazing to an old timer like me. Indeed, internets are wonderful things.

The focus of this paper was on the software consequences of some hardware features of the PowerPC 601, 604, and 620 chips when used in a multiprocessor configuration, particularly where test-and-set locking on a CISC based multiprocessor was about the same cost as a load or store instruction. On modern RISC machines, using a weakly-ordered memory access model, attempting a lock can require pipeline and/or cache flushes that cost between 50 and 200 cycles. The exact cost is technology dependent, and differs significantly between models of the PowerPC series.

This can have ramifications all the way up and down the chain. Even at the OS architecture level, the optimum point in the trade-off between having lots of little locks that must be taken frequently, but are not held for very long, versus having a few locks that are taken infrequently but are held for longer periods, now becomes different for different implementations. The approach of these two groups was to "keep it simple." Most programmers do not use complex locking schemes, so simple mechanisms are optimal no matter what the architecture. They also avoided locking by taking advantage where possible of atomic operations that did not require locks: using the "fetch and add" instruction for keeping statistics, and using "compare and swap" instructions for manipulation of singly linked lists. They encapsulated the complex and processor dependent parts of user-mode locking into two library routines, "`check_lock()`" and "`clear_lock()`".

They took an interesting approach to deadlock avoidance. Instead of using the traditional technique of global lock ordering, they used a "lint"-like static deadlock analyzer, called SDLA, which detects potential deadlocks by exploring the locking-scheme tree.

They implemented an "affinity scheduler" that takes account of the fact that a thread has a large amount of state information stored in the cache of the CPU it most recently occupied. Transferring the thread to another CPU requires moving all that state information, so they give threads an affinity for their most recent CPU.

Benchmarks of a quad-601 machine running at 75 MHz gave dynamic locking rates of 60-80 K locks/second. They got between 3.1 and 3.9 times the throughput of a single processor on their quad-601.

A Flash-Memory Based File System

by Atsuo Kawaguchi, Shingo Nishioka, and Hiroshi Motoda, Hitachi, Ltd.

Flash memories are interesting devices. They are nonvolatile. They are more rugged and faster than hard disks. They have no moving parts. They are cheaper and take less power than DRAMs, and they are just about as fast for reads. However, writing to them is a killer. They can't be overwritten simply, as one does with a DRAM. Not only DO they have to be erased first, you have to erase a full sector at a time – 64 KBytes for the devices described in this paper. Although erasing a sector takes but a second, you only get 100,000 of those erasures before the device wears out and has to be replaced! Designing a filesystem for these gadgets is a tall order.

The approach taken by the authors of this paper was to design a naive disk emulator that could be implemented at the device-driver level and that required no changes to any filesystem-level code. They borrowed some concepts from the log-structured filesystem, hoping to exploit the observation that if one knows which block is the next one to be written, one can erase it in advance of when it is needed.

They imposed a log structure at the device-driver level in support of a random access filesystem (Berkeley FFS) by interposing a layer of address translation in the driver. When a "write" command comes along, you allocate the next clean sector to it, and record that address in the pointer table. When a "read" command comes along, you use the pointer table to locate the sector that has the data. There are some interesting tricks in making sure that the pointer table (which is in RAM) can be reconstructed reliably after a system crash from just the information in the nonvolatile Flash memory. As usual, getting the sequence of operations exactly right is the key to success.

As with the LFS, the bottleneck is in the cleaner when the filesystem is almost full. Benchmarks show read performance similar to the BSD Memory filesystem (about 700 KBytes/sec) and write performance in the range of 150 KBytes/sec to 300 KBytes/sec, unless the cleaner gets into the critical path, at which point things run at cleaner speed, about 50 KBytes/sec.

In conclusion, they felt this approach was OK for a personal computing device, but not suitable for a server. They felt that to take better advantage of the Flash memory characteristics would require an integrated filesystem and device driver.

TRON: Process-Specific File Protection for the UNIX Operating System

by Andrew Berman, Virgil Bourassa, and Erik Selberg, University of Washington

TRON (named after the heroic fictional protection program from the 1982 Walt Disney movie "TRON") is a layer of file protection on top of the normal UNIX access control. It restricts access at the level of granularity of the individual process. You can use it to protect yourself from Trojan Horse attacks such as "the Malcontent's Home Page."

The user-visible parts of TRON are four new system calls: "`tron_fork`", "`tron_grant`", "`tron_revoke`", and "`tron_get_capability_list`".

`Tron_fork()` creates a new protection domain with capabilities that are a subset of the capabilities of the current domain, then forks a process running in the new domain. Plain old UNIX "`fork()`" and "`exec()`" do not change the protection domain, so non-TRON-aware programs get the behavior they expect. TRON capabilities are layered on top of UNIX file protection, so a process can be running as root and still have it's potential to do damage limited to a small set of files.

`Tron_grant()` and `tron_revoke()` are used to pass selected capabilities from a client process to a server process. `Tron_get_capability_list()` lets a process know the extent of its capabilities.

The kernel mods are limited mostly to vectoring system-calls through their corresponding TRON wrapper routines. If the process domain is the null domain, the wrapper routines do nothing and so are very fast. If you don't use TRON, you don't notice that it's there.

In summary, the advantages of the TRON approach are openness, flexibility, ease of use, and minimal overhead.

The slides of the talk should be available from Erik Selberg's home page at the University of Washington. *(http://www.cs.washington.edu/homes/speed)*

ILU/CORBA Inter-Language Unification

by Bill Janssen, Xerox PARC
Summarized by Peter H. Salus

CORBA, in case you didn't know, stands for Common Object Request Broker, of which the original architecture and specification were published at the end of 1991. ILU is Inter-Language Unification. ILU permits the use of different programming languages in a single program by providing an o-o module interface system that is language independent.

Janssen did a good job of describing the how and why of ILU. He used the construction of a simple calculator as a code example around which he built his explanation.

In brief, the various modules in ILU can reside in either the same or different address spaces. In fact, they can be in different spaces on different machines running different systems. It all appeared really neat.

They Come From Palo Alto

Summarized by Win Bent
<whb@ucs.usc.edu>

As session chair Phil Winterbottom pointed out, the three papers presented here were seamlessly united, at least in geographical origin. Just to make sure the point was not lost (and so that no attendees were confused about which session they were in), Jeff Mogul brought T-shirts for the speakers which said "We Came From Palo Alto." Needless to say, this was somewhat of a miscellany session.

SIFT – A Tool for Wide-Area Information Dissemination

by Tak W. Yan, and Hector Garcia-Molina, Stanford University

This talk presented some real advances in the challenging area of wading through the information-source flood. At Stanford, the authors have set up a system to look at Netnews articles and compare them with interest areas of some 13,000 subscriptions. Articles which are deemed interesting to a subscriber are then sent as mail. The criteria used to decide what's "interesting" are of two types: Boolean (if it's got the key word, send it), and vectors used to create a dot product (if the product of several key words exceeds the threshold, send it). The subscriber can then give "relevance feedback" to adjust the vector values. Another interesting approach they use is in the keyword indexing: rather than making an index of the articles and checking that against the subscriber profiles, they index the profiles and check the articles against that! This is a major savings in time and memory. Although the system scales well, using mail to deliver the results is a bottleneck, so the authors are exploring other means of presenting the output, such as gopher or the Web.

Performance Implications of Multiple Pointer Sizes

by Jeffrey C. Mogul, Joel F. Bartlett, Robert N. Mayo, and Amitabh Srivastava, Digital Equipment Corporation, Western Research Laboratory

Jeff Mogul once again arrived at USENIX with an interesting concept, a thorough study, and a low-key but fast-paced presentation. This time, he and his coauthors looked at a previously unstudied aspect of 64-bit computers: what effect do large pointers have on performance? Using some contrived worst-case programs and several real-world programs, they came up with an answer: it depends, but it's usually small. Their summary: both large and small pointers should be available to the programmer, just as large and small integers are available.

The data (and graphs) supporting this conclusion are interesting: for most of the programs studied, and for most reasonable-sized problems, using 64-bit pointers slowed execution by less than 5 percent. Generally, when the number of pointers outgrew the cache, performance began to drop off. This comes as no surprise, but now there are data to support and quantize the effect. This was an encouraging talk, and my guess is that it will be cited in DEC's Alpha AXP sales literature.

Idleness Is Not Sloth

by Richard Golding, Peter Bosch, Carl Staelin, Tim Sullivan, and John Wilkes, Hewlett-Packard Laboratories

Most computers sit idle for some periods of time, although not for lack of work to do. The authors of this paper have worked out a system to predict when such idle times will occur, how long they will last, and what jobs to schedule for those periods. Anyone can predict the future, you might point out, but these guys act on their predictions! Unfortunately, this talk suffered from the thorough study they gave the problem: they looked at many different predictor algorithms (timer-based, rate-based, etc.) and their accuracy (how well the predictor guessed the start and duration of an idle period), and presented a confusing graph in a low-key, vague-content talk. This was surprising and disappointing, given the potentially high value of the work.

Works-In-Progress

Presenters' summaries
Coordinated by Peg Schafer

Cicero, Software Distribution System

by Glenn P. Davis
<davis@unidata.ucar.edu>

For years we've been copying data from memory to files to memory again. `malloc()`, `write()`, `free()`; `malloc()`, `read()`, `free()`. I'm sick of it. Programmers writing "user level" code like databases have attempt to come up with efficient strategies, duplicating or thwarting work by the OS system designers in the virtual memory system. Boring. The problem is that we are handling data with a longer lifetime than the current program (file data) with different abstraction from the more transient data (memory). We really just need one abstraction, with controls for lifetime and sharing. With the availability of "mapped" files and standard locking interfaces, OS designers have provided us the mechanisms needed. The "arena" interface I describe is an attempt to provide an interface where persistent data can be manipulated using the programming idioms used for memory resident data. This WIP describes *Cicero*, a system for installing and managing software packages across a large heterogeneous network. This system is being developed at NASA's Langley Research Center (LaRC) in Hampton, Virginia by members of the Integrated Computing Environment (ICE) team. Although there are many different aspects of ICE, this paper focuses on the ICE team's work in progress concerning the issue of software distribution over a large installed base of clients.

SunScreen, A Graphical User Interface Builder for TK

by Stephen Uhler
<stephen.uhler@sun.com>

SunScreen is a prototype graphical user interface builder for TK that provides a direct manipulation interface for building, editing and testing TK applications. *SunScreen* learns about new interface elements (widgets) from extensions to TK, and automatically configures them as part of the interface builder. A novel dynamic "smart-grid" combines the familiarity of a spreadsheet-like WYSIWYG interface, with the powerful constraint-based geometry management capabilities of TK to provide automatic widget alignment, distribution, and resize management capabilities. Applications under construction may be tested from within the user interface builder.

Top version 3.4 and beyond

by William LeFebvre
<lefebvre@athens.dis.anl.gov>

Top 3.4 has just been released. I will summarize the changes to the program. I will also present some ideas I have for future versions and take (reasonable) suggestions from the audience.

Performance of Pseudoservers for the X Window System

by Ethan Solomita
<ethan@cs.columbia.edu>

A pseudoserver in this context is a proxy server that sits between the X application and the X server, serves to funnel all I/O between the two, and can manipulate the information. This is used for, among other things, sharing a single X application among multiple displays (e.g. workgroup computing).

The catch is that it requires an extra hop for all information transmitted between client and server. Most people would tend to assume that pseudoservers would make for a serious penalty in performance, but analysis showed that there were no sizeable delays for most interactive uses of an X application.

I think that this result might lend more credence to pseudoservers as a tool, where they might otherwise have come under suspicion.

WebRunner

by James Gosling
<jag@sun.com>

WebRunner is a browser that is extensible. It builds on the network browsing techniques established by Mosaic and expands them by adding dynamic behavior that transforms static documents into dynamic applications. Current documents in Mosaic are limited to text, illustrations, low-quality sounds and videos. *WebRunner* eliminates these limitations by adding the capability to add arbitrary behavior. Using *WebRunner* you can add applications that range from interactive science experiments in educational material, to games and specialized shopping applications. You can implement interactive advertising and customized newspapers. The possibilities are nearly endless.

In addition, *WebRunner* provides a way for users to access these applications in a new way. Software transparently migrates across the network. There is no such thing as "installing" software. It just comes when you need it (after, perhaps, asking you to pay for it). "Content" developers for the World Wide Web don't have to worry about whether or not some special piece of software is installed in a user's system, it just gets there automatically. This transparent acquiring of applications frees developers from the boundaries of the fixed media types like images and text and lets them do whatever they'd like.

VINO: An Operating System for Resource-Intensive Applications

by Christopher Small
<chris@das.harvard.edu>

What do applications want from the operating system? In the case of resource intensive applications such as multimedia, real-time, and database management systems, the answer is that they want it to get out of the way. Conventional operating systems are inflexible and uncooperative. Most policy decisions are fixed or limited to a few choices. Process scheduling, virtual memory, and buffer cache management are compiled into the kernel. Internal system facilities (e.g. synchronization primitives, the components of the file system) are not exported to use level. This forces applications to re-implement them in user space.

The *VINO* kernel, under development at Harvard University, is designed around three ideas: applications control resource management policy, the kernel structured as a collection of reusable tools, and all kernel resources share a common interface.

Allowing user processes to control kernel policy will permit the development of resource intensive applications with higher performance, less effort, and better integration with the operating system.

Indirect

by Karl Ramm
<karl@ac.duke.edu>

Indirect is a system for the secure delegation of authority and execution of system administration tasks via an insecure network, using Kerberos for security, and TCL for extensibility. *Indirect* consists of a TCL extension/client library and a server program incorporating a set of SafeTCL interpreters.

Supporting Distributed Systems over Low Bandwidth Networks

by Larry Huston CITI, University of Michigan
<lhuston@citi.umich.edu>

Network latency hampers the usability of distributed file systems over low bandwidth networks, so we have modified our disconnected AFS client to propagate file modifications asynchronously. We resolve cache misses and preserve cache consistency in the ordinary way, but do not incur any network latency on most file system operations, even mutating ones. Operating in this partially connected mode over a slow network significantly outperforms fully connected operation over a fast one.

Internet Information Commerce

by Nathaniel Borenstein, First Virtual Holdings, Inc.
Summarized by Jerry Peek
<jerry@ora.com>

I didn't hear about many technical breakthroughs in this session. The things that First Virtual Holdings do are almost obvious. But, I believe, no one else has done them first.

FV's system doesn't handle every type of commerce and transaction. It's a way to pay for information which can be downloaded over the Internet.

It's similar to the "shareware" concept of software: try before you buy, only pay if you want to. But it should yield much higher compliance rates than shareware because it makes payment automatic and penalizes those who consistently fail to pay.

Both the seller and customer do initial registration with FV over the Internet. Then they give confidential information (credit card or bank account numbers) off-line (and pay a small sign-up fee). Parties are registered and identified by their email addresses and a unique account identifier.

Sellers have a few ways to provide their product, including special software that FV provides (a patch kit for the WU ftpd, CGI scripts for Web servers, shell scripts and a C API for programmers). Customers use standard Internet software like telnet and Web browsers.

When a customer sees something she wants to buy, she gives her FV account identifier. That information goes to FV, who send an email message to the customer to confirm that she really does want to make the transaction. If she does, she sends a simple email reply to FV; FV charges her credit card and credits the seller's bank account (minus a small transaction fee). If the customer doesn't want to buy, she denies the transaction. If the customer didn't make the transaction that she's asked to confirm, the transaction is cancelled and her account ID is changed.

There's no encryption used over the Internet, so this model can work for any user in any country. All information that would help thieves is kept off the Internet.

FV's goal isn't to make a lot of money on the payment system, but rather on services that USE the payment system. Their payment system is operated on a cost-recovery basis.

They want to be leaders in intellectual property commerce. Specifications and details are freely available. Use anonymous FTP to *ftp.fv.com* and see */pub/docs*, email to *info@fv.com* for a canned reply, or point your Web browser to *http://www.fv.com.*

Libraries

Summarized by Win Bent
<whb@ucs.usc.edu>

Libckpt: Transparent Checkpointing under UNIX

by James S. Plank, Micah Beck, Gerry Kingsley, University of Tennessee, and Kai Li, Princeton University

The need to checkpoint a program, ensuring that intermediate results are saved, is a long-standing one. There is a large class of programs which require hours, days, or weeks to run, and the longer they run, the more anxious the user gets about the computer failing before the program completes! In a clear and engaging style, Jim Plank presented a library which he and the coauthors developed to simplify this task. With no changes to the source code, a programmer can get basic checkpointing, and with minor additions, the checkpoint size and time can be reduced, greatly improving efficiency. The authors have created a well-thought-out system including both existing and innovative techniques, and have paid good attention to the details.

Optimizing the Performance of Dynamically-Linked Programs

by W. Wilson Ho, Wei-Chau Chang, and Lilian H. Leung, Silicon Graphics Computer Systems

This talk, whose authors are "a boring group from a fascinating company" (SGI), addressed a common problem with dynamically-linked programs: when starting up, these programs have a high overhead due to resolving the dynamic links, and when running, there is another performance loss as the program resolves indirect memory addresses.

One fairly straightforward approach they used was compiler optimization: all modules in a program are examined and symbols resolved, so that formerly-dynamic addresses can be turned into static addresses. Another approach was to do some of the dynamic address resolution at compile time, resulting in what I would call "pseudo-static" linking. Using this method, the "true" dynamic linking only needs to happen if a library changes, such as when installing a new version. Another method, rearranging shared libraries to optimize memory usage, seemed to offer the least improvement. However, all three techniques showed that with a little (?) thought and some extra compile-time work, the speed of dynamically-linked programs can approach that of statically-linked ones.

DP: A Library for Building Portable, Reliable Distributed Applications

by David M. Arnow, Brooklyn College, CUNY

The goals and guidelines which David Arnow presented in this paper seemed, at the outset, to be conflicting: portable and flexible, yet low-level for maximal speed and efficiency which smacks of trying to get all three of Good, Fast, and Cheap. However, the results presented here looked good: an API (what we used to call a "library"), usable on various flavors of UNIX, which handles the tasks of managing and communicating between processes on a set of UDP-connected machines. One aspect of this system came as an afterthought: the mechanism is thoroughly reliable, although only if a single CPU is down. If a second goes down, the system "enters a strange state, then panics," but ensures that no wrong results occur.

There were two memorable moments I experienced during this talk. The first was the shock I felt when Arnow explained that he ran his distributed applications in the background on students's machines. If the situation at Brooklyn College is at all like the one here at USC, I feel sorry for both him and the students! The second moment was when he mentioned checkpointing, and said he was inspired by the talk on *Libckpt* earlier in the session. This, to me, is what USENIX conferences are all about, and why I keep coming back: because even the teachers can become the students.

Economics of the Internet

by Hal Varian, University of Michigan
Summarized by George W. Leach
<gwl1@gte.com>

Hal Varian, an economics professor from the University of Michigan, spoke on the Economics of the Internet. Unfortunately I felt that most of this talk was rather dry and often sounded like an Economics 101 class lecture.

The main thesis of this talk was that a pricing model must be devised to allow the Internet to continue to grow and become self sufficient. The talk covered four general areas:

- Pricing of transport

- Pricing of content

- Network externalities
- Intellectual properties

The pricing of the transport mechanisms for access to internet resources centered around the issues of quality of service and congestion control. Examples of quality of service include access speed, guarantees of deliver, security, etc. Different pricing models are needed to provide incentives to customers to choose a low quality of service as demand increases on the internet.

Content pricing for information must take into account the high fixed cost of development. However, there are no incremental costs involved in this type of product because there is virtually zero cost to the developer for copying. Varian drew comparisons with other industries such as air travel, software and telecommunications. These industries are highly competitive and often result in price wars. It is difficult to sustain a competitive marketplace in these industries and often the results is either a monopoly or oligopoly. A great deal of discussion followed regarding pricing models for these industries.

Regarding network externalities, the speaker discussed accounting and billing methods. The telephone company was mentioned as an example of centralized billing versus the postal system which is decentralized. Varian claimed that 80 percent of the cost of a phone call involved all of the book keeping necessary for billing. However, after the talk Andrew Hume of AT&T Bell Labs challenged his figure. Most billing is driven by regulations, which can vary from state to state.

The speaker ended by briefly discussing intellectual property. The current laws make enforcement costly. Shareware was pointed out as an example where sometimes the author can recoup intellectual property pricing in an effective manner. The entire area of intellectual property and pricing need to be rethought in the context of the internet.

File Systems

Summarized by Bryan Costales <bcx@lady.Berkeley.EDU>

File System Logging Versus Clustering: a Performance Comparison

by Margo Seltzer, Keith A. Smith, Harvard University; Hari Balakrishnan, Jacqueline Chang, Sara McMains, and Venkata Padmanabhan, University of California, Berkeley

Margo Seltzer presented a study contrasting the Log (LFS) and Fast (FFS) file systems of 4.4 BSD-Lite UNIX. Files were classified by size with those <64k considered small, those >1MB considered large, with those in between considered medium. Operations included read/write I/O, and meta-data operations like creation and deletion. According to Margo, "the bottom line is that meta-data is better under LFS than under FFS."

An interesting side-conclusion of her benchmarks was that FFS seemed to perform better when rotational delay was made zero.

During the question-answer period John Ousterhout filibustered, complaining about the apparent discrepancy between the 1993 and 1995 results. Margo pointed out that the benchmarks used in the two were very different, so results would differ.

Meta-data Logging in an NFS Server

by Uresh Vahalia, EMC Corporation, Cary G. Gray, Abilene Christian University, and Dennis Ting, EMC Corporation

Uresh began by referring to the previous talk, "It looks like everybody agrees that journaling file systems are the way to go, so I don't have to defend that." He then described a method for logging meta-data transactions on an NFS server. Essentially, some of the statelessness of NFS was given up to allow logging of creates, deletes, and the like. A small, slow disk held the log, with the only requirement that it be large enough to not wrap between cache flushes.

Logging dramatically improves a servers recovery time after a crash. Seconds, rather than minutes. This approach differs from that of the NFS Toaster in that it runs under UNIX and applies logging to an underlying BSD Fast File System.

Heuristic Cleaning Algorithms in Log-structured File Systems

by Trevor Blackwell, Jeffrey Harris, and Margo Seltzer, Harvard University

Trevor Blackwell presented the challenge (and one solution) to insuring that the I/O associated with LFS cleaning not interfere with normal file system activity. He proposed performing cleaning during periods of file system idle time, and that two seconds define the minimal idle time. Experience in commercial and university environments illustrated great success in minimizing the impact of LFS cleaning with this technique.

A trivial algorithm was developed to detect idle times and to trigger background cleaning. Trevor quipped, "A better title would be 'A Trivial Algorithm,' because I first thought algorithms would be complex, but turned out not to be."

Architecture

Summarized by Rasit Eskicioglu
<rasit@cs.uno.edu>

The New-Jersey Machine-Code Toolkit

by Norman Ramsey, Bell Communications Research, and Mary F. Fernandez, Princeton University

Mary talked about the New Jersey Machine-Code Toolkit, a software tool that helps programmers write applications that process machine code. Applications such as assemblers, compilers, and debuggers all have to work with machine instructions. Translating high-level language code into the machine-level representation (encoding) or back (decoding) is usually difficult and error prone. Traditionally, some applications avoid machine instructions by using the native assembly language. However, it is not always practical to use assemblers, as, for example, when adding instrumentation after code generation. The main goal of the Toolkit is to ease the development of such applications by automating the translation of the machine instructions. The Toolkit generates code from a set of specifications. The specification language supports both RISC and CISC architectures, and currently, specs for MIPS, SPARC, and Intel 486 are written.

The Toolkit has 3 parts: the Translator translates matching statements in high level language code into ordinary code, the Generator generates encoding and relocation procedures, and the Library implements the instructions and relocatable addresses which refer to locations within the instructions. Applications can use the Toolkit for encoding, decoding, or both. A disassembler, for example, takes a stream of instructions and decodes them using matching statements. A matching statement is a case-like statement where alternatives are labeled with patterns that match instructions or sequences of instructions.

The Toolkit's specification language uses pieces of instructions called tokens to describe machine specific details of an instruction in a given architecture. A token is broken into one or more contiguous ranges of bits, called fields. Opcodes of instructions are defined by patterns that are built from constraints on fields. Finally, a constructor – a simple function from operands to a pattern – defines an instruction.

The Toolkit's specification language bridges the gap between the stream-of-bits required by the machine and the symbolic representation desired by the user. Using these four basic concepts, complete instruction sets of both RISC and CISC architectures can be described.

Two applications were presented to show why the Toolkit is both practical and useful. One of the applications has to "recognize" instructions and the other has to "emit" instructions.

The first example is `ldb`, a retargetable debugger for ANSI C. As with the other debuggers, `ldb` analyzes the code to implement breakpoints. Basically, a breakpoint can be implemented by using flow analysis, i.e, looking at an instruction and figuring out which instruction to execute next. Usually, this flow-analysis algorithm is simply a large, nested-case statement. However, coding it would be tedious, and one can easily make mistakes. Instead, the Toolkit's Translator is used to generate this case statement automatically.

The benefits of using the Toolkit to build `ldb` were: first, it required very little code and the code was easy to understand, thus making it easy to retarget the `ldb` by simply writing the machine description; second, generating matching statements from the specifications has hidden the details and provided implicit error checking; third, the application was fast, as the Toolkit generated decoding code with the fewest number of tests.

The other application that uses the Toolkit is `mld,` a retargetable, optimizing linker for MIPS, SPARC, and Intel 386 architectures. Originally, `mld`'s code generators were designed to emit assembly code to simplify retargeting. However, it was too slow for link-time code generation because the assembler must be executed each time a program is linked with libraries.

The code generators of mld were modified to emit binary code directly. The emission of binary code was achieved by calling encoding library procedures of the Toolkit, which were generated by the Toolkit's Generator from the specs of different architectures. Similar to `ldb`, building `mld` with the Toolkit was easy and `mld` was 15% faster in emitting binary code.

The New Jersey Machine-Code Toolkit is available by anonymous FTP from *ftp.cs.princeton.edu* in directory *pub/toolkit*. Also, there is a homepage on the WWW as *http://www.cs.princeton.edu/software/toolkit.*

ATOM: A Flexible Interface for Building High Performance Program Analysis Tools

by Alan Eustace and Amitabh Srivastava, DEC Western Research Laboratory

Alan described a program instrumentation tool called ATOM (Analysis Tools with OM). ATOM is a very flexible and efficient code instrumentation interface for building high-performance analysis tools.

The motivation for developing ATOM was the ever increasing advances in technology that made today's computers and therefore program behavior very difficult to understand. For example, today's processors use multicaches, pipelines, multiple buffers, and multiple functional units. Based on these complexities, compilers use loop unrolling, code scheduling, and inlining. Finally, today's user applications are huge, and further complicated with synchronization, memory allocation, deallocation, and threads.

However, while computer technology was advancing rapidly, program analysis tools were not. One of the reasons behind this was the fact that every tool is written from scratch. Besides, much of the information required, for example, to build a memory analysis tool, is both confidential and hardware dependent, thus can only be used internally. Also, some tools, such as SPEC92 is program specific and cannot be used for every application. Some other tools are compiler or language specific, and some others require operating system modifications.

ATOM is a tool building system which is used to build new tools. It is flexible enough to build a variety of different tools for application programs, compiler groups, and chip designers. It is also fast enough to run on real applications, such as databases and CAD, and is both language and compiler independent.

ATOM is like a Trojan Horse. One can take an application and embed in it the tool-specific calls that analyze particular states inside the application. At the end of the program, a call to a tool-analysis routine would print out, as a side effect of its execution, some piece of information that wasn't known about the application.

Code instrumentation is not new. It dates back to Pixie, a MIPS instrumentation tool that modifies an executable, saves some register states, and outputs a few block counts. ATOM is a system with which one can easily build a tool like Pixie, and much more.

ATOM's programming model is composed of a group of procedures. Each procedure is composed of many basic blocks, and each basic block is composed of a set of instructions. ATOM allows the user to take a program and read it into an intermediate format which allows three types of primitive operations: navigation, information, and instrumentation. Navigation primitives allow users to move around. Information primitives provide static information about the program, its instructions, or its procedures. Instrumentation primitives allow users to add calls to analysis procedures before or after the instructions are analyzed. Alan later presented several simple examples to show how ATOM can be used to design program instrumentation tools.

The performance of an instrumented application is related to the number of places that are instrumented and the amount of work done at the instrumentation points. If the instrumentation is not done very often, then the application will run without any major performance penalties. However, it is important to note that the performance scales with the tool's complexity.

ATOM is effective in writing high-performance tools for several reasons. First, it allows selective instrumentation; for example, users can instrument only conditional branches. Second, its comprehensive data-analysis techniques allow users to reduce the frequency of analysis-procedure calls; for example, it determines identical blocks in the code. Finally, if it detects the loop invariant's value in a register, it moves the instrumentation out of the loop.

ATOM is now included in the OSF 3.0 release as an advanced development kit. An advanced version which uses shared libraries and executables is currently being tested. Thread support, improved instrumentation and runtime performance, and support for instrumenting the OSF kernel is also planned.

Adaptable Binary Programs

by Susan L. Graham, Steven Lucco, and Robert Wahbe, University of California, Berkeley

Robert's talk was about the implementation issues of program instrumentation tools. Program instrumentation allows the monitoring of the dynamic characteristics of programs at the lowest level. Instrumentation of a program mainly involves an analysis of the code and the insertion of meta-code to monitor various characteristics. When the instrumented program runs, the meta-code would then produce the desired information. The major goals of this work are robustness and efficiency. Robustness means that the transformation works with binaries produced by different compilers and on different platforms. Efficiency means that the transformed binary runs as efficiently as possible. Binary transformation applications include instruction level analysis tools, performance analysis tools, debugging tools, software translation tools.

Implementing one of the above tools basically consists of three steps: (1) analyze and disassemble the binary, (2) insert the meta-code, and (3) update the surrounding control addresses and allocate the registers for use by the meta-code. Implementation of these steps requires the knowledge of the location of the code in the binary, the control addresses to be updated, and the list of free registers. In general, current tools either use heuristics to derive the information or make conservative assumptions about program behavior. Usually, heuristics are very compiler dependent and might not work with every compiler. Also, programs composed of different modules that are written in different languages complicate building and testing binary transformation applications, while conservative assumptions incur significant runtime and potential-space overhead.

One approach is to have the compiler emit the minimum necessary information. This approach is very similar to asking for debugging information from the compiler. The necessary information is very simple to generate. The goals of this approach are to eliminate the need for heuristics and the need for conservative solutions.

Current systems face several problems. The first problem is related to the disassembly step. One common example of such a problem is the compiler intermixing the data with the code segment. For example, the compiler inserts the jump table of a case statement into the code segment. The heuristic approach to identify such data in the code segment is to search for the instruction sequence that implements the case statement and decode this sequence to find the location of the jump table. However, this approach would lead to problems of data being mistaken for code or code being mistaken for data.

The second problem is related to relocating the code. The control addresses must be updated after inserting the meta-code. This might not always be possible, especially, for example, when the address of the location in the jump table is computed dynamically.

One other problem is related to the allocation of registers for meta-code use. A trivial solution is to save the original contents of a few registers, use them in the meta-code, and restore the original contents of those registers. Fortunately, this solution is not always necessary as, for most programs, there are a few unused registers at most points in the program. However, current systems cannot easily determine the control flow of a program to look for such free registers.

Conservative solutions to the above problems include interpretation, dynamic disassembly, and out-of-line patching. Usually, however, these approaches only provide partial solutions. For example, out-of-line patching solves the relocation problem, but not the disassembly or register allocation problems.

A new technique has been developed in which the compiler emits the required information to support binary transformations. The information emitted by the compiler include control, relocation, and register usage information. A binary program that contains this information is called an adaptable binary.

Experiments have shown that the overhead of processing such information during transformation is minimal. For example, inserting NULL meta-code before every memory access instruction did not introduce any transformation overhead at all. Similarly, allocating registers at every memory access instruction introduced a low transformation overhead between 0% to 9%.

Currently, a prototype adaptable binary system which supports both ecoff and a.out formats is operational. The system uses modified gcc to output the minimum adaptable binary information.

USENIX Board Meeting Summary

by Ellie Young
<ellie@usenix.org>

Below is a summary of the actions taken at the regular quarterly meeting of the USENIX Board of Directors which convened in New Orleans, Louisiana on January 14, 1995.

Attendance: USENIX Board: Rick Adams, Eric Allman, Tom Christiansen, Dan Geer, Lori Grob, Andrew Hume, Stephen Johnson, Greg Rose.

USENIX Staff: Diane DeMartini, Judy DesHarnais, Dan Klein, Zanna Knight, Ellie Young.

Guests: Dan Appelman, Paul Evans, Mick Farmer, Jeff Haemer, Peter Honeyman, Evi Nemeth, Elizabeth Zwicky.

Bylaws & Policies Documents

There was discussion concerning the revised sets of both documents containing the changes recommended at the October board meeting, plus further revisions based on discussions with board members, legal counsel, and Young. Zwicky offered to revise the Special Technical Group section of the policies document concerning the removal of officers. After a few more recommendations for revisions, the Board voted unanimously to accept the modifications to both documents. The proposed changes will be mailed to the members via first class mail in March. Members will then have 30 days to send negative responses to the proposed changes to the Secretary of the Association, and if fewer than 25 percent of the Members object, the amendment will take effect. If 25 percent or more object, the amendment shall not take effect until the members have voted on rescinding the bylaw.

Symposium on Operating Systems Design & Implementation (OSDI)

Young was asked to contact prospective candidates to serve as program chair, as well as notifying the cosponsoring organizations about our plans to have a second one in the Fall of 1996.

Electronic Commerce Workshop

It was agreed to limit attendance at this workshop by requiring position papers for admittance.

LISA '96 Conference

Rose and Hume will serve on a subcommittee with some SAGE Board members to evaluate proposals to chair the LISA '96 conference (see p. 29 in this newsletter).

CitySpace

It was agreed to once again sponsor the CitySpace project with a contribution of $7,500. [*Note: This will fund CitySpace's participation in the Multimedia Playground'95: Digital Intersections, which is an exhibition of innovative digital media projects at the San Francisco Exploratorium. As part of this exhibition, the CitySpace project team will conduct the sixth in a series of 3D-modeling and networking workshops. These events combine an onsite workshop with an Internet-wide call for participation utilizing the Exploratorium's T1 network connection. Working closely with a team of kids onsite, a general invitation to participate in the event circulates widely on educators' mailing lists, KidsNet, and to schools, museums, science centers, and afterschool programs across the country. On site, the CitySpace project will collaborate with the Exploratorium's existing community outreach programs to present a 4 week program for San Francisco Bay Area. This program focuses on networking fundamentals and an introduction to 3D visualization and graphics, and involves several underserved schools and community groups.*]

Phil Zimmerman's Defense Fund

There was discussion about whether or not the Association could donate to this (i.e., would such a contribution jeopardize our nonprofit status as a 501 (c) 3 organization), and also whether our membership is interested in contributing. It was decided to investigate the legalities of this, and also poll the membership at the Open Board Meeting (see article on p. 4 of this newsletter).

Next Meeting

It will be held in Dallas, Texas on March 17, 1995.

Computing Systems

by Peter H. Salus, Managing Editor
<phs@net.com>

With a new Editor in Chief, Dave Presotto of AT&T Bell Labs, *Computing Systems* again encourages submissions (large and small) from those of you with something to say.

As Dave writes in his first editorial, "I want to avoid focusing on any small area. I want *CS* to be both eclectic and practical. If it's built, it's useful, and it's not trivial or obvious, write it up. No gedanken experiments or vaporware, please."

Submissions (preferably in PostScript, (La)TeX, or troff (any macro set) should be sent to:

<peter@usenix.org>

Alternatively, four (4) copies hardcopy can be sent to:

Peter H. Salus
#3303 4 Longfellow Place
Boston, MA 02114 USA

In its first seven years, *CS* has published works as short as two pages and as long as 80.

C'mon. Expose yourself in public!

Worth Noting

by Zanna Knight
<zanna@usenix.org>

Electronic Commerce Workshop

Everyone is talking about doing business on the Internet, but few feel comfortable doing it. How will these urgent technical issues be resolved? If you are working on any of the technical aspects of electronic commerce, you may want to present your work at this workshop where your peers will gather to debate and discuss these issues. Attendance for the first two days is limited to 100, and attendees must submit a formal position paper. Four half-day tutorials, open to one and all, follow the first two days of the workshop. See page 56 for details.

San Diego 1996

Remember, if you haven't heard this already, USENIX offers an annual conference where we cover the hot topics in advanced computing. The dates are January 22-26. This is the one of the few opportunities to meet your peers. USENIX's only 1996 conference will have a special focus on operating systems practice and experience. If you haven't received the poster version of the call for papers, turn to page 58 for the details.

UNIX Security

If you haven't already received your registration brochure for this symposium, you will any day. In addition to discussion on the latest security topics, UniForum has put together a special security track, headed up by Jim Schindler of Hewlett-Packard. Topics include Confidentiality, Integrity, Implementation and Disaster Recovery.

COOTS

The program has not been finalized at press time, but look for topics such as Oberon, the C++ Standard Library, Fresco, plus more. The program will be available shortly after you receive this. Read *comp.org.usenix* for the complete program. Or, go to our Web site. The URL is *http://www.usenix.org.*

LISA '95

Are you an experienced systems administrator who can't find professional development seminars that match your skill level? Check out the new offering, a day on Advanced Topics in System Administration at LISA 95. The focus will be on late-breaking technical issues submitted by participants. Attendance is limited, and based on the acceptance of a position paper. So, if there's a technical issue you want to discuss, this is the place to do it. See page 61.

Mobile

. . . Last call. If you want to learn more about the technological advances in mobile computing, it is not too late to register for the Mobile Computing Symposium. It's the place to meet the top practitioners and researchers in this field. See the program on page 60.

SANS IV

It may not be too late to register for this conference. New courses include Managing the World Wide Web and the Twelve Best Commercial Tools. For more information, see page 63.

USACO Seeks Sponsors

by Rob Kolstad
<kolstad@bsdi.com>

The annual USA Computer Olympiad is seeking more sponsors. USACO features three rounds of computer programming contests for 9-12th grades throughout the USA. The final round is a week-long programming camp for the top 16 finishers of the first two rounds. The four finalists travel to the world programming champions known as the International Olympiad of Informatics (analogous to the Physics and Math Olympiads).

Like all Olympiads, the USACO's goals include: providing students with opportunities to sharpen their skills (enabling them to compete at the international level), enhance the quality of computer education, and select the US team for the international Olympiad.

The total cost of the three rounds and airfare to the IOI sum to almost $30,000 each year (the IOI host country pays in-country expenses for participants at the IOI). All moneys support the program – no fees or honoraria are paid to the coaches and sponsors.

As USACO chief of staff, one of my jobs is to assist in fund-raising. If you or your company would like to join USENIX in sponsoring USACO (with concomitant acknowledgments and association with excellence, opportunity, and the future), please contact me, Rob Kolstad, at 719-593-9445 or *<kolstad@bsdi.com>*.

SAGE, the System Administrators Guild, is dedicated to the advancement and recognition of system administration as a profession. In just two years, SAGE's membership has increased steadily, and there is growing recognition of SAGE as a representative in system administration issues. SAGE brings together system and network administrators for:

- professional and technical development,
- sharing of problems and solutions,
- communicating with users, management, and vendors on system administration topics.

SAGE News Editor

- Tina Darmohray
  <tmd@usenix.org>

SAGE Board of Directors

- Elizabeth D. Zwicky, President
  <zwicky@usenix.org>
- Hal Miller, Secretary
  <halm@usenix.org>
- Pat Wilson, Treasurer
  <paw@usenix.org>
- Kim Carney
  <kim@usenix.org>
- Paul Evans
  <ple@usenix.org>
- Bryan McDonald
  <bigmac@usenix.org>
- Paul Moriarty
  <pmm@usenix.org>

SAGE Working Groups

Group	Chair
sage-certify	Arch Mott
sage-edu	Ron Hall
sage-ethics	Ed Gould
sage-jobs	Tina Darmohray
sage-locals	Rene Gobeyn
sage-online	Mark Verber
sage-policies	Lee Damon

You can contact these groups via email at
<their-name@usenix.org> for example,
<sage-certify@usenix.org>.

SAGE Discussion Groups

sage-security
sage-managers
sage-outreach
sage-pt
sage-solo

SAGE Online Services

Email server:
majordomo@usenix.org

FTP server:
ftp.sage.usenix.org

WWW URL:
http://www.sage.usenix.org

SAGE Supporting Member
Enterprise Systems Management Corp.

SAGE
THE SYSTEM ADMINISTRATORS GUILD
NEWS

Thanks to Bryan!

by Rob Kolstad
<kolstad@usenix.org>

After bringing *;login:*'s SAGE section from infancy in July of 1992 to a full-fledged set of contributions, Bryan McDonald (*<bigmac@erg.sri.com>*) is taking leave of the SAGE editorship to pursue the rest of his life, which includes continuing service to USENIX and SAGE as a member of the SAGE board of directors.

I join the rest of the USENIX staff and SAGE Board in wishing Bryan best of luck as he balances a life full of enough commitments for two people.

Replacing him as SAGE editor is Tina Darmohray, long time SAGE member, support, and nowadays a security consultant. Please ship all your SAGE articles to Tina at *<tmd@usenix.org>*.

Welcome aboard, Tina!

I Dare You

by Tina M. Darmohray
<tmd@charleston.GreatCircle.COM>

In 1991 I gave a paper at the San Diego LISA V conference. The paper was about configuring sendmail. I like sendmail, as most folks have heard me confess, but it turns out that a lot of folks don't share my enthusiasm for the sport. Happily, there are a lot of other things to tell about that paper I gave at LISA V, that have very little to do with sendmail, so I'll proceed along those lines.

In the 1991 time frame, I was a dedicated attendee of the BayLISA meetings. Indeed, I was so enthusiastic about those meetings that I frequently gathered up a carpool of corianders to make the monthly round-trip trek of about 60 miles, from Livermore to the Silicon Valley to mingle with the other sysadmins.

The BayLISA meetings always begin with "announcements." Since Elizabeth Zwicky was the program chair for LISA V, she would stand up at the beginning of each meeting and announce the upcoming conference and encourage folks to submit papers for the conference. One evening, she encouraged even the neophyte by assuring us that we could successfully submit a paper. She added that if it was determined that what we had submitted sounded interesting, but that our writing skills were (ahem) not, that the program committee would see to it that we were skillfully mentored through the process to ensure that we produced a successful paper. (And I gazed fervently out the window at the sunset during her announcement.)

The next day at work, one of my coworkers, who had been persuaded to join the carpool the previous evening, arrived in my office with an agenda. He basically dared me to submit a paper for LISA V. I responded with, "Yeah? What would I write about?" "How about your sendmail stuff?" "Oh, go on; I've never written a paper. Besides, I don't even know what all the previous LISA papers look like." "Well, I ordered all the previous conference proceedings; here they are! How about it?" (He was prepared; I gotta give him that.) "Harumph. – OK, I'll call Elizabeth (gulp) and tell her what I want to write about (and show you that she's going to laugh at me and then hang up)."

Moments later (with the challenger lurking in the doorway) . . .

"Hello?"

"Elizabeth?" I managed to spurt out.

"Yes."

"Um um um um . . . This is Tina Darmohray and I heard you say at the meeting last night that you want us to write papers for LISA V. I looked at the list of topics that you have and the one that I might write about isn't on there, and so I know that you wouldn't want it, besides it is horrible, and, most importantly, my Comparative Literature Professor hated my writing; I can't write."Calmly, she responded,

"What topic do you have in mind?"

"Uh – sendmail!"

"You want to write about *sendmail*?" came the disbelieving response. And before I could get away, "Yes! That's a fine topic. We'll help you write it."

"It is? You will? OK. Thanks! Good-bye."

Then the agony of choosing a title began!

But, I did submit a paper, and it was accepted. Useful comments were returned from the review process, which I incorporated as best I could. Then it went off to Rob Kolstad for formatting for the proceedings. Rob still kids me about "the woman who wanted just one more change" in the formatting, as his printing deadline approached. So, I somehow came out with a wonderful friend through the experience, as well.

This year Paul Evans and I are the program cochairs for the LISA IX Conference. We'd like to emphasize that this conference is only as good as the papers and talks that are given at it. Most importantly, those papers are written by the folks in the trenches doing the work; that's the stuff that all of us want to hear about!

So even if you are like I was, convinced that I didn't have anything to say, and even if I did, I wouldn't be able to write it down, I'd like to encourage you, from my own experience, that the system administration community does want to hear about your work. (And, if you really can't write, we'll get a mentor to gently assist you.)

Request for Proposals to Chair the 10th Systems Administration Conference (LISA '96)

The USENIX Association and its Special Technical Group SAGE, the System Administrators Guild, are seeking proposals from people interested in chairing the 10th LISA conference, to be held September 30 - October 4, 1996 in Chicago, Illinois.

We are seeking an energetic person with the following qualifications:

- Knowledge of timely and appropriate topics in the field
- Ability to find and invite excellent speakers to submit papers, as well as to invite appropriate keynote and panel members
- Excellent administrative and planning skills
- Experience in the administration of large installations
- Good public speaking skills
- Excellent reputation in the field
- Attendance at previous LISA conferences (involvement in a previous program committee is helpful)
- Time to invest to insure success of the conference
- Ability to work together with USENIX staff, volunteers, and invited talk coordinators

Proposals should be brief (1 page) and should include the following:

- Statement of Purpose (such as why have another conference and how it can be improved)
- Form of submissions (such as abstracts, extended abstracts and/or full papers)
- Format (such as 3 days of technical sessions, panel sessions, etc)
- List of topics to be addressed (such as the call for papers)
- Any special features (such as plans for improving the quality of the papers)

- List of potential program committee members and/or a cochair. (Note that while most USENIX conferences have had an individual program chair, proposals requesting a cochair are welcome.)

- Biography and references

Proposal due date: April 14, 1995

The program chair is not responsible for the conference's Invited Talks Track, Tutorials, BOFs, Vendor Exhibits, and conference site selection/ logistics. However, the chair is expected to work very closely with the coordinators for the Tutorial and Invited Talks program. The selected program chair will be invited to attend the program committee meeting for the LISA '95 conference in mid-May '95. Please address all inquiries and proposals to the Association's Executive Director, Ellie Young <*ellie@usenix.org*>. Proposals will be evaluated and selection of a chair will be made by a subcommittee composed of USENIX and SAGE board members.

A Moving Experience

by Debby Hungerford, Octel Communications Corp.
<debby@octel.com>

Octel Communications Corp. moved to a new campus last year. My group (engineering system administration) was responsible for defining and installing parts of the new computing and network environment for engineering, and moving it to two out of five new campus buildings. This article describes some of things we learned in that move.

The engineering move was well organized across traditional corporate service groups and dedicated engineering services groups. The engineering system administrators worked with cross-functional groups, such as Information Services and Facilities. Octel makes voicemail systems, so we have unusual requirements for phone lines of various kinds. The telecom and lab support groups dealt with this, while the engineering system administrators focused with Facilities and IS on the engineering computing and network environment.

The Octel computing environment today has 450 engineering employees of which 43 are not supported locally. In August 1994 we moved approximately 350 engineering employees – we hadn't integrated VMX yet and have grown since.

The engineers primarily use Sun Workstations and PC's. We have over 700 data connections in engineering, with 640 hosts. A little less than half are Suns, a little more than half are PC's; a few MAC's and other types of connectivity appear, as well. When we moved in August 1994, our numbers were smaller, of course.

Any move is going to cost "something." An important up front decision is what costs are acceptable and whether you should take a conservative or liberal approach. The Octel Engineering computing and network environment is managed conservatively because there is no one place where we can bring everything to a halt in order to install a brand-new development environment. Octel's focus is on the voicemail systems themselves. More importantly, Octel's engineers require reliability and predictability as an absolute. On the other hand, a company like SGI builds the computers it must also use in Engineering.

This dictated a strategy of being extremely well organized and performing risk elimination. I also had to consider the current stage of evolution of the System Administration team – we had not gone through a major project together such as this and it was important to demonstrate to ourselves and our users, in a quantifiable way, what we had become.

Several key items contributed to our move being a success:

- We assessed our risks and minimized them before moving

- We made no changes to our environment during the move – all were done before or after the move

- We supplemented the system administration staff with interns, contractors, and knowledgeable users

- We made special arrangements with our service vendors and had Polaris on site to help deal with the older Sun equipment

- We ensured lots of spare hardware and cables were on hand

- We created "rescue" servers – Suns that could bring any Sun client or server back to life

- We documented all data connections before the move

- We pre-labelled equipment with the old-and-new location, and machine name

- We compiled detailed checklists, updating them daily as the move got close, and shared them with the entire system administration move team

- We assigned roles to teams where the leader had a solid technical level in the area specified, and partnered the leader with a less experienced person (i.e., printers, terminal servers, etc.)

- We provided a headquarters that took updates from staff (work teams) and kept running statistics on the status of our many systems (increased morale and improved focus). The headquarters team also took care of communication and administrative details:

- We worked in teams of two to reinstall and fix problems

- We supplied radios – one per work team at headquarters and with IS people

- We ensured that physical needs and creature comforts were provided. This included a regular flow of food, a break area, and hotel rooms for those who needed them

- We moved into a campus which was built for us. The new buildings/environment were designed to provide a good foundation for the present and the future

- Move coordinators for each of the engineering divisions helped the engineers get prepared before the move and were the focal point for all issues for two weeks after the move

- We developed a strong bond with the other groups with whom we were working

- We had terrific shirts made up for the move team. This helped visually identify responsible people during the move and heightened the team spirit

- Training was available for all engineering personnel who weren't needed for the move on the Thursday and Friday of the move

- During the first week of move-in, system administrators wandered the halls in order to provide easy pickings for engineers who needed help. We also increased our standard support hours and on-call coverage for two weeks

- Our system administration team was committed to making the move a success. We could see that the Information Services and Facilities teams were similarly committed.

Overall, we were perceived, by management and our users, as having done an exceptional job with the move. We received a lot of positive feedback, the environment came up very quickly and reliably, and the network layout came up very well. The team was commended by a critical path project for getting them up four days earlier (including the labs) than had been forecast.

In the area of assessing and minimizing risks, we planned for the following risks (shown here with their resolutions):

Risk: Sun 3 servers with SMD disks.
Resolution: Converted to SCSI disks except for mail server, and had a backup root disk ready to go, with Polaris doing the tear-down and bring-up.

Risk: Sun 3/80 workstations on engineers' desktops.
Resolution: Migrated all in offices (60 of them!) to Sparc 5's before the move.

Risk: Sparc workstations.
Resolution: The Sparc's were pretty reliable but we still put nine Sparc 5's aside as spares.

Risk: Auspex and non-mainstream equipment.
Resolution: Hired vendors to do tear-down and bring-up. There were no problems with the Auspex server, and the others issues were minimized by having the vendor deal with them.

Risk: Solbourne
Resolution: Simple machine, so we tore it down and brought it up ourselves, but had Solbourne on standby in case we needed them.

Risk: Jukebox
Resolution: Simple to tear down and bring up, but had vendor come in day after bring-up to help test it.

Risk: Network Viability.
Resolution: Took network hardware to new site night of shutdown, hooked it up and tested by stress-testing the networks.

Risk: Sun download ports (our engineers use them heavily).
Resolution: Had terminal server ready to be installed, and ended up using it.

Risk: Engineering users in a non-engineering building who shared the same Novell server as corporate users.
Resolution: We installed a dedicated server for these users before the move and moved the users and server intact to the new campus.

Risk: Users prepared their own systems for the move.
Resolution: Provided very detailed shutdown and packing instructions; offered demos for this on Sun's and PC's.
Risk: Data protection.

Resolution: Ensured a recent full dump had been performed, then performed incremental right before shutdown started. Warned PC users to do backups.

About a week after the move, the system administration move team got together for a post-move meeting. You've read what we thought was positive, now here's what we felt went wrong. I think you'll see from the items below that many of these were very fine ideas, but we didn't follow through in one area or another.

- We combined our Help Desks (system administration, IS, and Facilities) and provided a single campus-wide hotline for the move coordinators to call. This was a good idea, but we didn't develop an automatic way for the end user to get feedback about the call beyond the "request" system we use in system administration.

- The system administration group provided daily updates to the move coordinators on all calls assigned to us, but the move coordinators didn't know how to use the information until the second day when we sent them an explanation.

- The telecom group left a notice for users if their phone wasn't working by move-in day. That was nice, but there was no follow-up notice when it was working and we didn't cover bad data connections in the same way.

- A whiteboard was going to be set up in the lobby that would indicate what major areas were up or down and, if down, when they were expected to be up. This was a good idea, but the white board didn't show up!

- There were detailed move instructions that had a lot of good information from the Service Team (Facilities, IS and System Administration) but they were too thick to get through.

- The system administrators followed the movers to get equipment set up and running. To do this effectively, the movers were to tackle a building and floor at a time (there are four) but the move-in wasn't staged this way after all, so the system administrators were not as effective as they would have been.

- We provided "open me first" boxes for cables, keyboards and mice but some of our users didn't use them properly. This delayed setup.

- We had a good mapping of data connections before we moved, but we had to freeze it a week before the move so the changes that happened between the freeze and move created a lot of extra work.

- The movers were very good with the equipment going into the computer room, but we had to remind the movers to be gentle with lab disk drives.

- We requested that the movers not block the data jacks so that we could get data connectivity up quickly and without banging our heads, but they blocked the jacks.

- We had lost-and-found areas for each building and floor, but they were not organized or inventoried so they became dumping grounds.

- We had key users on site, especially for setting up labs and that was great. We should have had more.

- The system administration team received feedback about our attitude being very positive – with a smile. The negative side was that our users hit different system administrators with lower priority tasks and this made us less effective. We increased communicating with each other which helped.

- The system administrators are accustomed to doing data patching, and we did for changes and last minute patches which was good but IS is the keeper of patching information. We neglected to develop a coordinated plan with IS for doing and maintaining data cable patching until after the move.

After the campus move, VMX merged with Octel, which was another move and that's another story!

Summary of the Actions Taken at the 1994 SAGE Board of Directors Meetings

January 20

New officers for the coming year were elected: Zwicky – President, Evans – Secretary, Schafer – Treasurer. Simmons and Parseghian were thanked for their services in 1993 as President and Secretary.

The committee tasked with looking into our sponsoring regional LISA seminars was disbanded due to a lack of enthusiasm and of sufficient resources; further discussion of this topic will be taken up by the education working group.

Parseghian and Zwicky will work with the staff to develop themes for a SAGE calendar.

Parseghian and Moriarty formed a committee to investigate new members services.

Most of the defunct working groups were disbanded (e.g., conferences, pr, and awards), and the board liaisons will be responsible for making their groups conform to the charter. Schafer will work with the staff on producing a SAGE information bulletin board for the conferences.

It was decided that the SAGE mailing list would be restricted to members only and that the SAGE-announce mailing list will be available to all subscribers.

The Board offered its congratulations to SAGE-AU on its first anniversary, and will contact them about offering bulk discounts for their purchasing the SAGE jobs booklet.

It was suggested that the LISA conference organizers investigate adding a security component under the purview of the program chair.

Moriarty volunteered to handle the SAGE booth and BOF at the SANS conference.

March 1994

The working groups SAGE-conf, SAGE-robbies, and SAGE-pr were disbanded. SAGE-outreach and SAGE-pt were made into discussion groups.

Zwicky has lined up volunteers to staff the USENIX/SAGE booth at UniForum.

Local groups will be authorized to post their meeting and organizational announcements to the SAGE-announce and SAGE-locals mailing lists.

It was decided to develop a SAGE-jobs-offered mailing list and that Wilson will be responsible for monitoring the postings.

April 1994

Wilson and Moriarty formed a committee to get volunteers to organize SAGE BOFs at conferences.

It was decided to add a Members Services section to the SAGE Web page which will allow members to advertise products and services.

It was agreed to cover balloting fees for a limited number of SAGE members who wished to participate in standards.

June 1994

The Board expressed its interest in having the staff look at locations outside California in the early Fall time slot when selecting sites for future LISA conferences.

It was agreed to reappoint Bryan McDonald as SAGE publications coordinator for the coming year.

It was decided that members of the nominating committee will not be eligible to run for office for the SAGE Board of Directors, and nominations will close after the LISA conference.

It was decided that SAGE would like to once again co-sponsor the SANS Conference, and Young will discuss the arrangement with its organizer, Alan Paller. Zwicky, Wilson, and Schafer offered to serve on the organizing committee.

It was decided to once again sponsor sys admin tutorials at the UniForum conference in 1995, and Zwicky will work with the staff on the program.

The subcommittee's recommendation to accept the proposal from Darmohray and Evans to co-chair the LISA 1995 conference was accepted.

It was decided that each new member will get a copy of the most recent previous SAGE publication, instead of waiting until a new publication is released.

It was decided there will be only one open Board meeting per year (at the LISA conference), and that a SAGE BOF will be scheduled at the USENIX annual conference.

Each board member was charged with calling at least five companies to solicit them to become supporting members.

July 1994

Brent Chapman was thanked for his services as SAGE postmaster, and the Association's sys admin, Scott Seebass, will take over this task.

It was agreed that the Secretary should post summaries of the minutes from the board meetings to the sage-announce mailing list.

All SAGE board candidates will be invited to attend a candidates forum at the LISA conference.

Zwicky and Wilson will work on revising the proposal submitted by Schafer on support of local groups.

It was decided that every SAGE member will receive a calendar, and further discussion concerning promotional items will be dealt with at the next meeting under budget discussion.

Evans was designated as the board liaison for the LISA '95 conference, and will keep everyone apprised of the program committee activity.

September 1994

Dinah McNutt and Pat Parseghian were thanked for their efforts in putting together the technical sessions and invited talks at the LISA Conference.

The staff was asked to look at Chicago and Atlanta as possible non-West Coast venues for future LISA conferences.

The awards committee recommended that Larry Wall be the recipient of the 1994 SAGE Outstanding Achievement Award.

It was reported that Ron Hall had agreed to serve as chair of the SAGE-edu working group.

It was recommended that SAGE-standards, SAGE-vendors, and SAGE-security working groups be disbanded. (SAGE-security was turned into a discussion group, and SAGE-vendors has not been disbanded pending receiving a new charter and chair).

It was decided to investigate the possibility of developing an on-line resource center which would consist of a forum for discussion as well as tools and techniques for system administrators.

It was decided to let the staff conduct negotiations with various people who wanted to translate the *SAGE Jobs Description* booklet into French and Swedish.

The budget for fiscal year 1995 was approved.

The suggestion to have Greg Rose serve as USENIX Board liaison (replacing Tom Christiansen) was approved.

It was decided that the results of the SAGE salary survey (which was being conducted at the LISA conference) will be mailed to SAGE members only, and that they also be published in *;login:* after the members have received them.

October 1994

The SAGE-publications working group was disbanded.

It was decided that due to lack of activity, corporate member solicitation be removed from the board's action item list.

Young will work on putting together a list of items for inclusion in a SAGE policies document, since bylaws are not needed.

December 1994

Board members were encouraged to contribute and also look for authors to submit articles, reviews, and columns for the SAGE news section of *;login:*.

Early Computer Quotes

The following is from the business section of *The Kansas City Star*, January 17, 1995:

"Computers in the future may weigh no more than 1.5 tons."
– *Popular Mechanics*, forecasting the relentless march of science, 1949.

"I think there is a world market for maybe five computers."
– Thomas Watson, chairman of IBM, 1943.

"I have traveled the length and breadth of this country and talked with the best people, and I can assure you that data processing is a fad that won't last out the year."
– The editor in charge of business books for Prentice Hall, 1957.

"But what . . . is it good for?"
– Engineer at the Advanced Computing Systems Division of IBM, 1968, commenting on the microchip.

"There is no reason anyone would want a computer in their home."
– Ken Olson, president, chairman and founder of Digital Equipment Corp., 1977.

Report on the IEEE Mobile Computing Systems and Applications Workshop

by M. Satyanarayanan
<M_Satya@MOZART.CODA.CS.CMU.EDU>

Introduction

The goal of this two-day meeting was to foster interaction among active workers in mobile computing, with a view toward cross-fertilization of ideas. Given the youth of the field, such interactions could have substantial impact on its future direction. In keeping with this goal, the conference organizers chose to have a small, informal workshop rather than a larger and more formal conference. The workshop was sponsored by the IEEE Computer Society Technical Committee on Operating Systems, in cooperation with ACM SIGOPS and USENIX.

The workshop was held on Thursday and Friday, December 8-9, 1994 at the Dream Inn in Santa Cruz, CA. The weather was beautiful and the cyanosed locale spectacular – alas, it is not clear whether this helped or hindered the workshop, since many longing looks were evident on the faces of participants as they gazed out of the windows! The General Chair, Darrell Long, had done an excellent job of selecting the workshop site and setting the stage for the workshop. He was assisted in local arrangements by two student volunteers, Jim Cummiskey and Chane Fullmer.

What follows is a summary of the discussions that took place during the workshop. It is based on notes taken by four student volunteers (Peter Grillo, C.K. Toh, Adrian Friday, and N. Asokan). They did an excellent job of taking detailed and complete notes. Any errors or omissions in this document are certainly my responsibility, not theirs.

This digest is intended to be a supplement to the papers in the proceedings, not a substitute. Rather than producing a verbatim transcript, I've tried to focus on those interactions that seemed most insightful, controversial or evoked most response from the audience. Such a report must, by its very nature, be subjective. I've tried to be as objective as possible, but I'm sure there are places where my personal biases show through. My apologies in advance if you attended the workshop and your favorite comment, question or discussion isn't mentioned here.

Thursday, December 8

Models and Methodology

The theme of the opening session, chaired by Randy Katz, was the identification of novel ways of thinking about mobile computing and using these viewpoints to derive system structures. Doug Terry of Xerox PARC presented the first paper, on the architecture of the Bayou system. The problem addressed by this work is the maintenance of consistency in shared, replicated data reposito-

Obtaining the IEEE Proceedings

Copies of the full proceedings of this workshop will be available from the IEEE Computer Society after late March 1995. Its complete title is "Proceedings of the Workshop on Mobile Computing Systems and Applications," and its order number is PR06345. The publisher can be contacted via email at *cs.book@computer.org*, via FAX at (714)-821-4641, and via phone at 1-(800)-CS-BOOKS.

Computing Systems

To get a complete list of contents, and to order back issues of the Association's journal, contact MIT Press. Back issues of the journal are available at the following prices: $9 for individuals and $18 for institutions. Outside the US add $5 per issue for postage and handling. Canadians add an additional 7% GST.

The MIT Press Journals
55 Hayward Street
Cambridge, MA 02142
1 617 253 2889
<journals-orders@mit.edu>

SAGE Jobs Booklet:

Short Topics in System Administration Series: Job Descriptions for System Administrators can be ordered for:

$5.00 for SAGE members
$7.50 for nonmembers

$3.50 overseas postage is added to orders outside the US and Canada.

Contact the USENIX office at 510 528 8649 or send email to *<office@usenix.org>*

ries updated by mobile hosts. Bayou's model of consistency is reminiscent of that of Grapevine, built nearly fifteen years ago.

Updates by a mobile host to a particular repository site are tentative, until those updates are received by the primary site responsible for the data in question. Updates are propagated to all data sites in an epidemic, or rumor-mongering fashion and may become visible to other mobile hosts even before final validation by the primary site. Since secondary data sites may not receive all updates in the same order that the primary site finally chooses to order them, the state of the data on secondary sites may differ and tentative updates that have already been applied may have to be rolled back and reapplied after other incoming updates. The system is careful, however, always to make it clear to users which data is derived from tentative updates and which from permanent.

The questions after the presentation addressed two areas: clarification of the consistency guarantees, and the rate of convergence. Doug indicated that a tentative update may be rolled back and reapplied at a given secondary site at any time until that site has heard what the final "commit" ordering is that the primary site has chosen for the update. He also indicated that the anti-entropy mechanism responsible for update propagation may be executed many times for each update (if there are many servers). But the update procedure converges as long as no host remains partitioned forever.

The second talk, by Arup Mukherjee, made the case that existing work on mobility focused on computation and communication, to the exclusion of control. His thesis was that a rich taxonomy of applications emerges when control is given due prominence, and that the taxonomy offers valuable insights into structuring applications to function effectively under the constraints of mobility.

In particular, Class 7 applications (in his 7-element taxonomy) were currently under-represented, but offered many advantages in a mobile environment. The questions after the talk addressed two issues. One question was whether real applications could be mapped as cleanly into the taxonomy as the speaker claimed. Arup replied that complex applications are often composed of subsystems in classes distinct from that of the parent. He added that it is the task of the system builder to examine an application at a level of granularity relevant to the issues being considered. The other question was really an observation that Class 7 applications demand mobile hosts to have substantial computing resources; something like an Infopad will not suffice.

Perhaps the most controversial item in the workshop was the talk entitled "Are Disks in the Air Just Pie in the Sky?," given by Mike Franklin. The approach is to use a network as a rotating information medium by periodically retransmitting the entire contents of databases. The central idea behind this work is to superimpose multiple disks spinning at different speeds on the broadcast medium in order to support nonuniform data access. Rather than fetching data on demand, clients continuously listen to the transmissions and cache information of interest to them. This approach is especially valuable when the network has asymmetric bandwidth.

The flurry of questions at the end of the talk covered many aspects of the work. Satya pointed out that networks in mobile environments tend to be unreliable: how can you depend on broadcast data when mobile? Mike agreed that this was a problem, but that it could be addressed by prefetching critical data. Mary Baker observed that broadcast may not be supported by some mobile networks. Karin Petersen warned that receiving data costs energy; it is therefore an illusion to believe that the broadcast approach comes for free. Mike stated that for some important applications, such as advanced traffic information systems, battery power is not a concern, and that given sufficient demand, there is no reason why lower-power mechanisms for monitoring the broadcast could not be developed.

File Systems

Lily Mummert presented the first talk in this session, chaired by Peter Honeyman. Her talk focused on techniques to cope with the performance and reliability of mobile networks. The techniques spanned three areas: deferring update propagation during periods of low bandwidth, opportunistically using high bandwidth when available, and using an abstraction called "dynamic sets" to reduce network latency during search. In the question period, Peter Honeyman asked how log replay is actually performed during trickle discharge. Lily answered that the replay occurs as a set of iterations on small parts of the log. Terri Watson pointed out that applications had to be changed in order to use dynamic sets, and that they have to be able to tolerate the reordering of requests implicit in the use of dynamic sets.

The second talk, on shrinking a replay log using peephole optimization in a postprocessing step, was presented by Larry Huston of the Little Work project. This approach is in contrast to that of Coda, which applies optimizations incrementally. The primary advantage of the Little Work approach is that the optimization code is a separable component; hence, it is easy to apply to multiple file systems. Peter Honeyman asked what timestamps files received. Larry replied that they received the replay time, rather than the true modification time, because this allows programs like "make" to work correctly.

Jay Kistler asked what the asymptotic performance complexity of this optimization technique was. Larry answered that it was O(N**2) worst case, but that the running time in practice was quite acceptable. In response to a question from Terri Watson, Larry said that operation reordering was essential for up to 60% of the optimizations they were able to achieve.

Predictive caching was the topic of the third talk in this session, presented by Geoff Kuenning of the Ficus project at UCLA. The goal of this work is to reduce the burden on users of specifying files to be hoarded in anticipation of disconnection. The system uses a list of observed file references and a set of clustering algorithms to construct a plausible mapping of those references into distinct tasks. Hoarding is then performed on tasks rather than individual files.

In the question period, Lily Mummert pointed out that multiprogramming would complicate the clustering analysis, since the observed stream of file references would be the union of two or more distinct tasks. Geoff agreed that this was the case, but said that clustering analysis could be refined to distinguish between one primary task and a number of secondary ones, a common scenario in single-user multiprogramming environments. Jay Kistler asked how much simulation of the proposed scheme had been performed. Geoff replied that he preferred results from real use to simulation results.

Wiring the Campus

In this first panel of the workshop, moderator Rich Wolff began by observing that the title of the panel was only a loose characterization of the work represented in it. Each of the participants then gave a brief summary of their work.

Abhaya Asthana described the design of a shopping environment with wireless connectivity for each shopping cart. Vince Russo gave an overview of the deployment of a wireless network at Purdue University, using an ATM backbone switch to cope with *en masse* movement of many users, such as will occur between classes. Mary Baker reported on a new project, called MosquitoNet, to increase connectivity when switching a host between wired and wireless communication on and around the Stanford campus. Since all three projects are at a very early stage, there were no war stories to report. The ensuing discussion focused on two major issues, both relating to the campus wireless projects.

The first issue was whether truly "mobile" computing, in the sense of people computing while walking across campus, was either likely or desirable. Many members of the audience felt that a more likely scenario involved students using their portable computers in each classroom, library, etc., but not while they were walking. For this scenario, all one needs are network outlets at each desk in a classroom; wireless coverage is not necessary. Satya pointed out, however, that truly mobile applications do exist. For example, experiments are in progress at UC Santa Barbara to enable visually handicapped people to navigate on campus using portable computers to sense current location and to give directions with voice synthesis.

The second issue was the impact of campus mobile computing on social mores and etiquette. For example, how does one prevent electronic cheating such as by students passing zephyr messages to each other during an exam? Even with a perfectly honest population, there are issues such as whether it is acceptable for a person with a noisy keyboard to disrupt a lecture, or to intrude upon a discussion.

Application Frameworks

In the session after lunch, chaired by Dan Duchamp, four papers were presented. Each of these papers focused on a broad class of mobile applications, and described a paradigm or a set of techniques applicable to that class.

The paper on teleporting, presented by Frazer Bennett, reported on experience using a system that enables the display of an application to follow a user around as he moves, leaving program execution at the original site. This ability is especially convenient when combined with an active badge system that tracks user location.

Questions from Peter Honeyman and James Kempf probed the limitations of this approach. In particular, they were concerned that hiding display changes from applications would render the applications unable to adapt correctly to changes in display size or color characteristics. Frazer agreed that this approach would indeed be inadvisable for applications that were tightly coupled to specific display characteristics. Dan Duchamp asked how ambiguities, such as the presence of two displays in the same room, were resolved. Frazer replied that the user is iterated through the choices of display and can pick one. In response to a question from Karin Petersen, Frazer said that it is not possible at present to allow selective movement of windows. There were also a flurry of questions and heated discussion on issues of privacy and security.

The next talk, by Roy Want, described work at Xerox PARC on making ParcTab applications sensitive to the current physical location of the user. David Steere asked what the security consequences of losing a Tab were. Roy and Karin Petersen explained that each ParcTab was associated with a user, and that loss of a Tab was as serious as losing a key, though some additional security could be provided via a PIN code. Doug Terry added that the privileges of a Tab could be easily revoked by killing the proxy server associated with it. Randy Katz requested details of the infrared communication mechanism used by the Tabs. Roy said that the typical bandwidth was 19.2Kbps, although bandwidths up to 1Mb/s were possible.

Terri Watson then described her experience with designing applications for wireless computing. The theme of her talk was that developers should exploit application-specific knowledge to address mobile resource constraints. In certain cases, it is desirable to offer alternative actions to the user, allowing them to make performance versus cost decisions. Geoff Kuenning asked whether it is realistic to expect all existing applications to be rewritten according to this philosophy. Terri replied that the highest payoff applications would be rewritten regardless of effort involved, and that the total number of viable applications in a mobile environment were limited.

The final talk in the session was by Kenjiro Cho, reporting on the use of group communication primitives for mobile computing. The talk closely followed the paper, with emphasis on establishing that the performance overhead of this approach was indeed acceptable. In response to David Steere's question about behavior during network partitions, Kenjiro explained that ISIS only supports group communication in the majority partition. C.Toh asked whether clients needed to explicitly select a new primary server during partitions; Kenjiro replied that this selection was subsumed by ISIS.

Exploiting Mobility Commercially

Many participants have told me that this panel, representing industry's perspective on mobile computing, was the most exciting part of the workshop. Amal Shaheen of IBM Austin, the moderator of the panel, posed four questions for the panelists:

1. Is there money to be made in mobile computing?
2. What are the characteristics of successful mobile applications?
3. What is the impact of mobility?
4. What are the merits of a client-only approach versus one that requires modifications to both clients and servers?

She was confident that there is a lot of money to be made in mobile hardware, but felt that there is no data to decide whether the same is true of software. The trick will be to find out what the users expect and deliver something more than that expectation. She felt that packaging and ease of use were important characteristics of a successful application, *Lotus Notes* being a good example.Transparency can only go so far: things like conflicts and cache misses during disconnections are impossible to hide.

Finally, Amal observed that it is logistically much simpler to provide support entirely at the client-end. Server-end changes render existing servers incompatible, and are thus much less attractive. This remains true even when server changes offer substantial functionality or performance gains.

Murray Mazer (now at the OSF Research Institute) spoke next, and reported on his experience with mobile computing at Digital Equipment. He observed that a broad range of people in the computer industry (ranging from Bill Gates and market analysts to real users) believe that there is a market for mobile computing. He therefore believes that there is definitely money to be made in it.

He then pointed out that mobility will not be the differentiating factor in the future; rather, it will be the norm. Exactly when this will happen depends on when the infrastructure for mobility becomes widespread. Regarding applications, Murray observed that users are intolerant of bad interfaces. They will not go through poor interfaces to get to the cute functionality as we implementors might. They hate poor performance and unannounced missing functionality. Hence, we should strive to make the user-visible components easy to use; this, in turn, requires us to manage complexity in applications and services.

He expressed the belief that people will pay for valuable functionality; for example, cellular phones are popular even though their use is expensive. Rather than focusing on vertical applications, as in today's market, he suggested that remote information access was going to be the fastest growing and key class of applications.

Finally, Murray argued for making quality of service more explicit in applications: be more careful in setting user expectations, and as far as possible allow users to make explicit trade-offs of cost and performance.

The third panelist, Bill Fitler of Lotus, reported on his experience with the *CcMail* and *Notes* products. He first pointed out that there is definitely money to be made in mobile computing, and that the popularity of these two products is proof. He emphasized that total transparency is never going to be possible, and that users are not expecting it anyway. Mobility results in a very harsh environment for applications, and they often fail in serious ways under these circumstances. Bill also noted that support for mobility is much like support for fault tolerance: it has to be built-in and cannot be added on later.

Dorota Huizinga was the next panelist, speaking on behalf of herself and her collaborator Ken Heflinger of AST Research. She began by noting that their work had been inspired by Coda, and that they had persisted in their efforts to implement disconnected operation in DOS in spite of the fact that their measurements of write-sharing in the AST environment were significantly higher than those reported for Coda.

For the same reasons that the previous panelists had cited, their work was an entirely client-side implementation with no server changes. Dorota noted that many of the implementation challenges they faced had nothing to do with mobility; rather, they were caused by the memory addressing limitations of DOS. Finally, she expressed the belief that there was money to be made in mobile computing, but admitted that she was unable to substantiate this belief with specific data.

The talk of the next panelist, James Kempf of Sun Microsystems, was very brief. His primary message was that mobile computing applications would benefit greatly from widespread support for a special language that would allow applications to download code easily. In response to Peter Honeyman's prompt about *Telescript*, he agreed that the language should not be proprietary.

The last panelist, Bob O'Hara from Microsoft, was confident that there was money to be made in mobile computing. He observed that there were three portable computers in his presence right there at the workshop: a laptop, a pager, and a watch which was a joint product of Microsoft and Timex that could download his schedule from software running on a PC.

Peter Honeyman asked whether we are likely to see body implants, to which Bob replied that it didn't matter whether the hardware was worn on the outside or the inside. Bob was of the opinion that transparency was important because it was the key to allowing third party software developers to write applications easily. Barry Leiner asked how he hoped to hide limitations of the network for applications like video, to which Bob replied that he had not given this class of applications serious thought. On the matter of mobile applications, Bob observed that vertically integrated applications like appointment books tended to be the most successful.

The rest of the panel session consisted of a number of discussions spanning the range of topics touched upon by the panelists. Amal, Bill Fitler, and Satya engaged in a heated discussion about the level of abstraction at which support for mobility should be provided. Amal argued that the support should be at the file system level, because all applications could benefit from it. Bill countered that providing the support at a higher level (such as the *Lotus Notes* application) allowed more information to be used for conflict resolution. Satya pointed out that this need not be an "either/or" situation: Coda provides support at the file system level, but allows application-specific resolvers to be transparently invoked upon detection of a conflict.

A second topic of discussion was on the issue of usability. Satya observed that the harder one works to mask the ugly characteristics of a mobile environment, the more difficult it is to explain to naive users what has gone wrong when the masking is no longer feasible. The panelists agreed that this is indeed a difficult problem. Murray Mazer and Bill Fitler gave simple examples of how errors can be presented to users in meaningful and easily-understood ways, but everyone agreed that these merely scratch the surface of a difficult problem.

Marvin Theimer warned panelists not to place so much trust in marketing surveys. After all, pen-based computing had been predicted to be a major market but there are no signs of it taking off yet. He then offered the opinion that entertainment (including games such as multi-user *Doom*) would be the driving force of mobile computing. If this turns out to be true, he observed, the entertainment industry might pay for the cost of the mobile infrastructure. There was substantial disagreement on this conjecture. Many among the audience and panelists considered it unlikely that entertainment would pave the way for other mobile computing applications.

Dan Duchamp directed the panelists' attention to a different topic: academic research on mobile computing has focused on UNIX, while industry is almost exclusively focused on Windows/DOS. Dan asked whether this was a healthy dichotomy, and whether academic research should switch to Windows/DOS. Bob O'Hara replied that the industry approach could be characterized as "small steps for small minds." With the passage of time, the Windows family is getting to be more like UNIX. Further, visitors from universities do contribute their UNIX biases to industry. Hence Bob advised academia against giving up on UNIX, but not to forget about desktop systems such as Windows.

The last few minutes of the panel session were spent on a potpourri of topics ranging from cellular telephones to a revisit of the importance of entertainment. But the long day and the aroma of hors d'oeuvres from the next room sapped the vigor of the discussions. The panel and the day ended on a quiet note.

Thursday Evening: Exhibits

A set of exhibits from industry and universities, organized by Peter Honeyman, was displayed concurrently with the reception at the end of the first day. Peter had done an admirable job of ensuring that the exhibits were not mere marketing glitter but had something insightful to offer to the participants of the workshop. There were six exhibits, of which two were commercial products and four were research prototypes. Each is briefly described below.

IBM Mobile FileSync

Amal Shaheen and Tom Porcaro of IBM Austin demonstrated a new IBM product, *Mobile FileSync*, that has been bundled with Lanserver 4.0 for OS/2. Inspired by Coda, but differing considerably in its detailed design, this product supports disconnected file access in OS/2. The support is entirely at the client end, with no changes required to existing servers. The current version of *Mobile FileSync* provides support for hoarding, as well as for step-by-step reintegration via an interactive process. An important aspect of the implementation is that it is layered entirely above the file system switch. As a result, the support for disconnected operation works with any file system below the switch. The exhibit involved two IBM *ThinkPad* laptops on an infrared wireless LAN.

Lotus Notes and CCMail Mobile

These two popular products from Lotus were demonstrated by Bill Fitler. They are both examples of vertically integrated applications that originated in LAN networks but have been extended to mobile environments.

Notes is relevant to mobile computing because of its replication model. A client can connect to the network and obtain a replica from a server. Once a replica is downloaded, it can be used "off-line" (i.e., disconnected from its server). Considerable effort is made to hide whether you are on-line or off-line, but user control is possible via a sequence of menus. There is a full scripting language for creating filters, so that only desired information is collected from the server in any given connection.

CMail Mobile looks identical to the LAN version, with the addition of one new menu which deals with all the mobile aspects. This enables a user to send, receive and move messages back and forth between a mobile client and a server. The system allows you to set up default usage locations, and to associate those locations with attributes such as modem type and dial prefixes.

A variety of communication mechanisms, including over 150 modem types, is supported. These can be tried in order, rolling over from one to the next to discover a communication mechanism that works at the current location. Scheduling functions exist to allow the user to contact the server at startup, closedown or user-specified intervals. Filters can be constructed to select messages based on criteria such as size and priority.

PARC Tab

Norman Adams from Xerox PARC demonstrated the *PARC Tab* hardware that has been used in a variety of experimental projects. The *Tab* has a small, graphics-capable screen, 128KB of memory, and an infrared transceiver. The infrastructure at PARC consists of room-sized cells equipped with infrared transceivers. Each Tab has a server process running on its behalf on a workstation on the wired network. Applications on a Tab can be implemented as Tcl scripts that are executed on the server, or as stand alone programs with surrogate processes on the server.

The demo consisted of two cells and a *Sparcstation* functioning as server. The concept illustrated by the demo was that of "proximate selection." One example consists of a user walking into a cell, and selecting "forward call" on his *Tab*: his phone calls are automatically forwarded to the room he is in. Another example consists of an application to list available printers, with nearest first: when the user walks to a different room, the display automatically changes.

Wit

Wit is a research prototype built by Terri Watson of the University of Washington. The client hardware consists of infrared transceivers developed for the Xerox PARC *Tab* project and stock HP 1000LX palmtops. Largely unmodified PARC *Tab* code implements low-level transport.

The software system consists of two components: a network-side proxy and a palmtop system that extends the DOS environment to support multiple active applications through windowing, user threads, and network connections. `Tcl` interpreters in both components serve as the primary application programming interface. Application functionality is partitioned between the proxy and palmtop by dynamically defining and executing new `Tcl` functions on the remote side, with the goal of reducing both bandwidth consumption and user-perceived latency. Terri observed that `Tcl` treats all data as strings. This can complicate applications' use of non-ASCII data.

Marine Maintenance Assistant

Arup Mukherjee, of the *VuMan* project at Carnegie Mellon University, demonstrated a wearable computer. It consisted of a small computer with a *Private Eye* display and a hand held controller with three buttons. The software on the machine was customized for a specific application, that of access to documentation for maintenance tasks. The demonstrated version of the system was 80C186-based, but later versions of the system will be 386-based.

Teleporting

Frazer Bennett showed a brief video to illustrate his earlier talk on Teleporting. The video showed people wandering around, pressing their active badge buttons and having X-displays migrate to their current location.

Friday, December 9

Networks & Protocols

The first session of the second day was devoted to the topic of networking and protocol issues in mobile computing. Ramon Caceres chaired the session, in lieu of Krishan Sabnani who was unable to attend due to a personal emergency.

In the first paper, Raj Yavatkar examined the problem of end-to-end TCP adaptation in mobile environments. He observed that such communication often involves a short wireless segment and a much longer LAN or WAN segment. Standard TCP code fails to recognize the very different reliability characteristics of these two segments, resulting in unsatisfactory performance. Raj described a solution in which intermediary code allows TCP to independently adapt to the characteristics of the two segments. His solution provides substantially better performance, while preserving complete upward compatibility with existing clients and servers.

Barry Leiner asked whether the goal of not changing TCP was a valid one. TCP was designed with certain link-level characteristics in mind, and if those cannot be met, it is better to redesign TCP. Raj disagreed, saying that preserving TCP unchanged as far as possible has enormous practical value. Further, the 10-12% packet loss, typical of wireless segments, is far too high; improving link-level reliability is essential. Later, Barry asked whether it was appropriate to consider the proposed scheme end-to-end, because the intermediate code violates the end-to-end reliability semantics of TCP. The session chair shared the same concern and seconded Barry's comment. Raj replied that the situation was no different from that of a gateway.

Nigel Davies presented the second paper, describing experience with a mobile application for the electric utilities in the UK. The goal is to help linemen collaborate effectively with each other and with control rooms. The system developed for this requires each lineman to have a laptop with support for wide-area wireless communication. The client software includes collaborative tools for displaying and editing maps and provides users with feedback on the quality of the underlying communications network. Initial trials with the system have been conducted, and wider deployment is expected.

In response to a question from Mary Baker, Nigel said that the main feedback from users was that they wanted the client software to look and feel more like Windows. Users also wanted the collaboration software to better distinguish input from different users. Barry Leiner asked where information about network quality was obtained, and whether the TCP stack was bypassed in doing so. Nigel replied that TCP was not used, and that the custom-built RPC layer provided an interface for applications to obtain information about network quality.

The third paper in this session addressed the problem of wireless communication between mobile hosts in locations where there are no base stations or other mobile infrastructure. Dave Johnson described a protocol in which the hosts themselves serve as forwarding agents and thus constitute an impromptu mobile infrastructure. There was heated discussion over whether a user would like his machine's cycles to be used for routing someone else's packets. Dave observed that this was the price of membership in an ad hoc mobile network. Terri Watson asked if signal strength could be modified under program control; Dave replied that current wireless hardware does not permit this.

Finally, Allen Lao of UC Berkeley presented a paper on a video transport protocol for wireless networks. The novel feature of this protocol is its ability to dynamically adapt the bandwidth required to the current content of the video. Specifically, video segments with a large amount of motion can be rendered in a lossy manner without noticeable degradation of picture quality. Allen noted that this is the opposite of MPEG, where segments with lots of motion tend to result in higher bandwidth requirements. Jim Kempf asked how this worked with video conferencing, where lip sync is important. Allen replied that this could be handled by using low resolution for the mouth area, but ensuring that it was sampled frequently. The talk ended with a brief video demonstrating the concepts.

Accessing the World-Wide Web

Over a short period of time, the World-Wide Web has acquired star status as an information repository. This session, chaired by Jay Kistler, focused on the topic of accessing the Web from mobile clients.

The session began with Joel Bartlett describing his experience implementing a Web browser on an Apple *Newton*, communicating via a low-bandwidth wireless link. The talk was accompanied by a video demonstration. Joel observed that his strategy of partitioning applications, so that the CPU-intensive processing occurred on powerful servers, was critical to good performance. A German team that had also done a PDA implementation of *Mosaic* had gotten only 10% of Joel's performance. Terri Watson pointed out that Joel's strategy was consistent with the design philosophy she had espoused earlier in the workshop. Frans Kaashoek inquired about prefetching, and Joel replied that the next PDA screen was prefetched. Jim Kempf and Bob O'Hara asked about the client-server protocol and the server hardware. Joel replied that the protocol was custom-designed, and that the server was a DEC 5000 with a MIPS R3000 processor.

The second talk, by Josh Tauber, described a very different approach to mobile access of the Web. Web documents are now programs in Tcl/Tk that are executed at the client by an interpreter that enforces safety. At present the system works on IBM *ThinkPad* clients and *Sparcstation* servers over a 2 Mb/s WaveLAN wireless link.

Terri Watson questioned the basic assumption of the approach: namely, that authors would be willing to write programs rather than documents. She observed that a major reason for the success of the Web was the simplicity of the HTML format. Josh replied that authoring tools would help in this process, and that development of such tools was essential for the success of this approach. Terri then expressed skepticism about the portability of the approach: different versions of each document would be necessary to allow different types of clients and interpreters. Karin Petersen agreed, saying that translation between HTML and the client side filter was necessary; this would obviate the need for authors to foresee all possible client configurations. Josh observed that interface discovery techniques could be used to help. Murray Mazer suggested that for every type of mobile entity, there be a server agent that could perform appropriate translation. Finally, Marvin Theimer proposed that attention be focused on defining a standard PDA interface, rather than supporting heterogeneity.

In the final paper of this session, Geoff Voelker described a publish/subscribe approach to contextual behavior in Web documents. In this approach, active documents subscribe to some variables; these variables are periodically updated by agents. A change in a subscribed variable causes a document to be reloaded on a client. Josh Tauber asked how this would scale, since every time a variable changes, the corresponding agent has to inform all subscribers. Geoff replied that the work done at Xerox by Schilit and Theimer on using multicast to limit update traffic was relevant here. Darrell Long asked whether subscription is transitive, and was told that it was not. There was an extended debate about the meaning of "go back to the previous document," when the information used to generate the previous document might no longer be available. Jim Cummiskey observed that the previous document would still be in the local cache. Finally, in response to a question from David Steere, Geoff said that not much thought had yet been given to the issue of security.

Privacy & Anonymity

The first session after lunch was a panel on the topic of privacy and anonymity, chaired by Marvin Theimer. One of the panelists, Amir Herzberg, was unable to attend due a personal emergency.

The two panelists in attendance, Didier Samfat and N. Asokan, had independently addressed the same problem: that of ensuring the privacy and anonymity of a mobile user when he is far from the certifying agents he normally uses. Both approaches are based on using public key encryption to design authentication protocols in such a way that the mobile user's identity is not revealed to unintended parties. Didier presented a taxonomy of anonymity requirements and presented a solution based on the use of one-time aliases in authentication protocols in place of the user's real identity. Asokan advocated the notion of "limited disclosure of information" (regarding the user's real identity) to obtain practical anonymity.

The first important question raised was "is this a relevant issue?" Amal Shaheen asked whether it is even desirable to provide anonymity. Asokan replied that it is a policy issue and the goal is to be able to provide the mechanisms necessary to make anonymity possible if it is desired.

Another question debated at length involved the reliability of such an approach. A failed home site, or intermediary, would leave the mobile user with no means to obtain services. Asokan responded that such a failure was no different from a store's attempt to validate a VISA card today. In addition, Didier observed that the standard reliability measures, such as duplication of servers within a domain, are taken to ensure that such essential services are highly available.

Josh Tauber asked who would pay for the cost of establishing and maintaining intermediaries. Didier replied that this would most likely be done by businesses, but may also involve customer payment. There was then an extended discussion about encryption, and the need to make its use more widespread.

The final topic of discussion involved possible abuse of location information. Marvin Theimer gave the example of a person whose path regularly goes past a pornographic store. While various conclusions may be drawn from this raw data, an entirely innocent explanation is possible: the person may merely work next door. Many other similar examples were discussed by the panelists and the audience. After several examples of how location information can be abused, a majority of the audience was convinced that protecting privacy is indeed important. The examples revealed that mobility increases the possibility of abuse in two distinct ways: first, by permitting the perpetrators to work unsupervised in remote locations; second, by providing new types of information that can be abused. For example, if an insurance company obtains the cellular phone records of a customer, it may be able to determine that he often exceeds the speed limit and should therefore have his rates increased.

Panel: Agenda for Developers & Researchers

The final part of the workshop offered an opportunity for each participant to reflect on what he or she had heard over the previous sessions, and to brainstorm with a small group on four questions:

- Where would you like the field of mobile computing to be in five years?
- What can individual researchers do to influence the field?
- What can industry do to make mobile computing profitable?
- What are the three most important problems (technical or otherwise) to be solved for mobile computing to advance?

The participants were partitioned into five groups. Each group had a leader, whose primary responsibilities were to facilitate discussion and to bring the group back in time for the final panel session. The groups and their leaders were: "seal" (Terri Watson), "otter" (Murray Mazer), "whale" (Bob O'Hara), "dolphin" (Mary Baker), and "sea lion" (Doug Terry). The groups had about an hour and a half to brainstorm, and many of them chose to hold their breakout meeting outdoors. After the breakout session, the participants reconvened and one member of each group reported its conclusions in the final panel session.

Ken Heflinger, representing "seal" group, spoke first. His group believes that in five years there would powerful but affordable PDAs and that we will be living in a predominantly paperless world. The main thing researchers can do to influence mobile computing is to figure out how to make key technologies cheap, to work on fundamental technologies, and to combine things in useful and interesting ways. Regarding profitability, this group believes that using entertainment to whet people's appetites and exploiting advertising on PDAs as a source of revenue were two promising paths to paying for the mobile infrastructure. The three critical problems to be solved are perceived as: (a) realizing the global infrastructure (wireless everywhere, at low cost), (b) the development of software to deal with heterogeneity, and (c) battery power limitations.

The next panelist was David Steere, representing the "otter" group. David said that his group was divided into two camps: Peter Honeyman (whose views are too colorful to be mentioned in a respectable publication!), and everyone else (whose views are reported here). The group feels that vertical applications (such as the one described by Nigel Davies earlier in the workshop) and mobile infrastructure will be pervasive in five years. The main things individuals can do to influence the field will be to help develop the infrastructure, demonstrate the feasibility of mobile applications, and help understand consumer needs. The perceived obstacles to commercial exploitation of mobility are: the development of vertical applications, the need to provide interoperability across a wide range of platforms, and the need for a wireless communication infrastructure.

Bob O'Hara then summarized the deliberations of the "whale" group. In five years, this group believes that the global wireless infrastructure will be deployed, that the killer application for mobile computing will have been discovered, and that many different mobile devices and gadgets will exist. Individual researchers can help influence the field by trying to use and deploy new applications; university-based researchers are viewed to be particularly well-positioned to contribute to the development of wireless testbeds. This group drew a blank on the issue of profitability. The three big obstacles foreseen by this group are: (a) imprecise disconnection semantics, (b) absence of a ubiquitous infrastructure, and (c) running out of radio spectrum (especially for small companies that can't bid high at FCC auctions).

One of the group members, Barbara Liskov, made two additional observations. First, she observed that the problems of mobile computing are specializations of the problems that researchers have been addressing for many years in distributed computing. Second, she observed that mobile computing may require substantial revision to the fundamental primitives of distributed computing, such as RPC.

The next panelist was Barry Leiner of the "dolphin" group. In five years, this group expects mobile hardware to have advanced to the point where a "desktop in a pocket" is a reality – this will enable full sized screens and keyboards to be effectively "rolled up" for portability. But little global improvement is expected in the area of networks: they are expected to be sporadically available, of variable bandwidth, reliability, and heterogeneity. Integrated information access, the perceived "killer application" for mobile computing, will be pervasive.

The group felt that the most effective way for individuals to influence the field will be via prototypes that open users' eyes to new possibilities. Barry reiterated the point made earlier by Barbara Liskov, that many of the problems of mobile computing are really refinements of problems already encountered in distributed computing. He also reported that his group believes that a cooperative approach, like the Internet, was the best way for industry to build an affordable mobile infrastructure and thus maximize profits. Finally, the three most pressing problems in mobile computing are perceived to be (a) power management (b) scale-related issues and (c) user-perceived complexity in dealing with enriched service abstractions in a resource-poor context.

The last panelist to speak was Jim Rees of the "sea lion" group. In five years, this group expects wide coverage via high-bandwidth wireless communication, electronic commerce, and interoperability via open services. The group feels that the most effective way for individuals to influence the field is by developing better abstractions and metaphors for mobile adaptability, and by obtaining a better understanding of trade-offs. Regarding the question of profitability, the group has four suggestions. First, exploit cheap hardware. Second, treat mobility as a premium and charge more for mobile services. Third, ensure easy access to mobile computing facilities. And, finally, support electronic commerce in mobile environments. In the opinion of this group, the biggest challenges facing mobile computing are: (a) the absence of a mobile infrastructure and facilities for billing (b) the need for adaptability and (c) the need for consistency (so that the mobile world is not drastically different from the desktop world).

After the reports by the panelists, the floor was opened for general discussion. Three major topics emerged. One discussion, involving John Saldanha, David Steere, Karin Petersen and Mary Baker, focused on the question of what kind of devices it makes sense to render mobile. No real consensus emerged.

Another discussion, involving Barry Leiner, Doug Terry, Peter Honeyman and Amal Shaheen, explored the claim that mobile computing is merely a special case of distributed computing. The consensus that developed is that many of the problems of mobile computing are indeed subsumed by distributed computing; but there are important differences. For example, location transparency is often a goal in distributed computing, whereas location awareness is a requirement in many mobile applications.

The third discussion, between David Steere and Barry Leiner, examined the role of the entertainment industry in mobile computing. David expressed the view that we should bet on game manufacturers, rather than computer manufacturers as the driving force behind mobility. Barry disagreed, and said that history shows that the entertainment industry is a follower, not a leader. But the industry's ability to create a giant market for communication and cheap hardware can be exploited to advantage.

Final Thoughts

I have received feedback from many participants indicating that the workshop was quite a success. Many attendees felt that they had learned a lot at the workshop, and that they had been able to contribute effectively to it. They were especially appreciative of the informal format, the small size of the audience, and the quality of the presentations and discussions. They confirmed that many thought-provoking discussions and ideas arose during the workshop. Quite a few of them inquired whether there would be a follow-up workshop in a year or two.

Success does not, of course, come by accident. Many people worked hard behind the scenes to ensure it. Crucial to success were the efforts of my colleagues on the program committee: Dan Duchamp, Peter Honeyman, Randy Katz, Jay Kistler, Krishan Sabnani, Amal Shaheen, Marvin Theimer, and Rich Wolff. They did an excellent job of reviewing and selecting papers on a tight schedule. They also did a great job of chairing the workshop sessions – keeping things moving on time, and encouraging discussions. Darrell Long, the General Chair, and the other organizers (Richard Golding, Peter Honeyman and Luis-Felipe Cabrera) must also be complimented for their efforts in putting together a high-quality event. My secretary, Marge Profeta, helped me in numerous aspects of the workshop. But, in the final analysis, it was the level of participation and enthusiasm exhibited by the attendees that made this such a productive and enjoyable workshop.

What Good is a Gig?

by Scott Hazen Mueller
<scott@zorch.sf-bay.org>

I was telling my wife about the gigabit RAMs that are due out in the year 2000, and she asked me, "What can you do with a gigabyte of memory? Especially if you can only use 640K at a time?"

Now maybe that 640K limit will soon be as relevant as hand cranks on cars, but the question still stands. What sorts of things become possible? What becomes easy? What becomes more difficult? Furthermore, what about gigabytes of disk? Gigapixels of display? And gigabits of bandwidth? What will they enable in the future?

If the only differences new technologies bring are in terms of quantity, then I'm not interested. I don't care that much about running window systems faster, or having 100 applications resident at the same time, or whatever. Likewise, a 32k x 32k pixel display, while marginally mind-boggling, is not all that exciting if it's just one big screen that I run my Window system on. The future needs to be about new functionality, if it is going to be fun and interesting, and more importantly, if computers are to really penetrate society and become useful tools for everyone. New technologies should be used to unleash new capabilities, not just make the same old ones run faster.

Take, for example, the user interface. The most common user interface, the teletype, basically dates back to middle of the century. Some work has been done in the area of pen-based input, but that is a niche market and is likely to stay that way. Speech input has been around for a few years, but the current implementations have many drawbacks. This is definitely an area in which technological advances can make a difference. As computer power becomes cheaper, it becomes more reasonable to use brute-force algorithms, such as really large lookup tables, to decode speech input. I think that speech input will have a definite place in the user interface of the future.

The keyboard will likely be with us for some time to come. It is just too hard to beat a keyboard for bulk entry of text. Various firms are already doing work in new methods of attaching the keyboard. Remote control units for consumer electronics are already so ubiquitous that there are firms making programmable units that can replace several device-specific units. Converge these two trends, and you come up with remote keyboards, portable units that interface via infrared or digital radio to computers. I would like one right now, so I can do away with that annoying cable.

What about output methods? Larger memories are one enabling technology for larger displays, but I seriously doubt that we'll actually see monolithic gigapixel displays anytime soon. What I do consider possible is the multi-headed paradigm, with several physical devices sharing one virtual display space. That display space will contain more than just computer applications. Consider for a moment the effects of the convergence of television, telephony, and computing. AT&T hopes to make the television into a device to access information via the telephone. Computer companies want to integrate telephone and television access into your desktop system. Cable television companies want to start delivering telephone service over your lines, and give you information services over those same lines.

One way to look at this is as a conflict, in which one model (and side) wins, and the other falls by the wayside. Another way to look at this is to focus on the convergence, the melding of the three technologies. If you take a system with gigabits of bandwidth, several gigabytes of secondary storage, and a few gigabytes of physical memory, there is no reason whatsoever it cannot fill all roles at once.

Taking that premise, it begins to make sense to view physical display devices as windows into a private virtual information world, all sharing a common space, but from different points of view. Looking at it that way, then any one of multiple screens can be used in any of the roles, as a viewing device for text and graphical information (computer), as a point-to-point communication channel (video phone), or as a terminal for pre-programmed video (television).

Think of the virtual world as a sort-of virtual desktop. Instead of just a bunch of glass teletypes, the world-scape can have live video, several kinds of communications gateways, local applications, and who knows what else, all existing in parallel, and viewable from any display on the system. Instead of having a screen saver that has to monitor activity and then become active, have a virtual fish-tank somewhere in the world, and have the displays pan to it when they are not being used for something else.

Also, the portable keyboards I mentioned above could be used in conjunction with any of the display devices. If you want to work in a different room, simply carry your keyboard to a new place, pan the display to the area you were working in, and there you are. If the keyboards use digital radio to interface to the computer, the system could even track you as you moved around the building, and have your workspace ready for you.

What are some of the possibilities that are enabled by massive local storage? Right now, I can buy about half a gigabyte for a little over $200. That's a nice amount, but it's quite easy to fill that up just with a fully-featured system, not to mention the applications and interfaces I've been talking about. Even halving the cost per megabyte is not going to make much difference; a gigabyte of disk just isn't that much.

However, the cost will probably drop around 10x by the end of the century. That may well be enough to make a difference. For example, my wife's vinyl record collection takes about 11 linear feet of space. My first-order estimate is that it constitutes 10 gigabytes of audio data. At today's prices, it would run about $3500 to buy enough disk space to digitize all of that data. In a few years, the cost would fall to $350, certainly well within the reach of a computer-literate middle-class couple. At that point, not only does massive local storage become practical, but it also becomes desirable, because you can do things with the data on-line that you simply cannot do in the original format. For example, we could categorize every song from the collection by artist, title, type, enjoyment factor, date, mood, or whatever. Then, we could arrange to play them by any of those categories, at any time, without having to shuffle media. On top of that, we could buy new songs as they came out and add them to the collection, at a low incremental cost.

It probably wouldn't be practical to do the same with our videotape library but we could certainly reprocess them to newer media, e.g. 8mm or 4mm tape, in digital format, and archive them for future use. Instead of needing a dedicated device (the VCR) to view them, we could just load the archive tape into a drive used also for routine backups, and view them on any of the system displays.

I can't see handling our books in this way; they have physical properties that we find attractive. We do own plenty of paperbacks that frankly, I would archive, since they have a limited lifetime. For a lot of textual material, it would make sense to scan and archive it. I would do the same with our various paper records; even an 8mm tape takes much less room than a moderate-sized file.

There is a downside, of course, to having this much on-line capacity. It would be quite difficult to ensure good backups. This would create a tension between the distributed and centralized model. If the bandwidth from the providers could be put into place, it would become much more convenient to merely access data over the network, and let the provider worry about backup. However, if the cost for network access is too steep, then people will want more of their data on their own local system, so that they can access it for free.

In the final analysis, the communications channels are going to be the glue that holds everything together. Without sufficient bandwidth, video-on-demand becomes impractical. Without bandwidth as a driver, the convergence of television into the computer/telephone complex becomes much less likely. Without the convergence of television, I don't believe that the display technology will be driven into the shared mode I've envisioned. Without shared displays, the computer will be just a box that sits in the corner and processes text information at the paltry 112 kilobits/second available over ISDN.

The wide bandwidth channels will enable everything to happen all at once, and this is going to be the key factor in converging all of the current computer and communication technologies into one bundle. The networking will have to enable seamless interface between technologies. For example, my wife and I have two computers, two displays, keyboards and mice, two televisions in the house, and several telephone, including two in the office. All of these are basically unshared resources, even though we have the computers networked together. Tying the computers together at 10 Megabits/second enables some sharing, and we can exchange files and the like, but that is about all.

In order to share memory, I/O devices and peripherals, and generate the virtual world, it will be necessary to network systems at gigabit rates. Likewise, general-purpose video will require the same high data rates. Voice telephony does not require the high data rates, but establishing the in-building network makes it possible to attach phone sets as access points. Furthermore, if the phone sets are connected to the network, they can be used as voice-input terminals. For some functions this will most likely be more convenient than keyboard entry. As an example, a phone set in a family room could be used to order up video on the display (formerly television) there.

These technologies are exciting, and I look forward to seeing computers handle more and more communication tasks for everyone. What I really want to know is what sort of social changes these tools will bring about. Some changes are already here. There are many more great changes coming, and – like the person trying to envision the world today after seeing a car for the first time – what is visible now is just the very beginning.

An Update on Standards Relevant to USENIX Members

by Nicholas M. Stoughton
USENIX *Standards Report Editor*
<nick@usenix.org>

The following reports are published in this column:

- •ANDF BOF
- •NII/GII Information Infrastructure BOF
- •POSIX.0: Guide to POSIX OSE
- •POSIX.6: Security Extensions

Our Standards Report Editor, **Nick Stoughton**, welcomes dialogue between this column and you, the readers. Please send your comments to *<nick@usenix.org>*.

Snitch Reports

Not every article that is published in this column is met with total delight by all the readership. Now, that shouldn't be a surprise, but occasionally I receive some of the flames that result. That's part of my job, and I encourage you to let me know if something appears here with which you take offence.

The authors of the articles that appear here write on a voluntary basis (OK, USENIX pays the tab to take them out to dinner once in a while), and write as individuals, expressing a personal opinion. "Snitch" reports are not, ever, corporate opinions. POSIX, possibly more than most other International Standards groups, is driven by engineering principles rather than corporate politics. If you read this column regularly, then you should, if I'm doing my job right, get informed of progress (or the lack of it), of the reasoning behind apparently strange decisions, and of the overall direction of UNIX related standards. You should be told by those people in the thick of it. Of course, if the author's in the thick of it, and there have been two radically different approaches discussed, you might get to hear only one side of it, since the opinions are personal.

This column is largely factual, coloured by opinion, with a hint of controversy. I want it to be readable, not only by those who are in the midst of the action, but also, and mainly, by those around the world who cannot otherwise find out what's happening without waiting for the book to appear. If one of the authors of these reports says something you don't like, please submit a response for publication.

The IEEE process that we follow demands openness. Anyone can join the debate. The process demands consensus from all involved. If you disagree with the way that POSIX is going, then get involved in the debate. Of course, sometimes there has to be a losing side. If you're on the losing side, then you must simply accept the consensus opinion.

Report on ANDF BOF

Chuck Severance <crs@msu.edu> reports on the July 1994 meeting in Nashua, NH:

Background

Five years ago, I attended an OSF member meeting and went to a presentation on "Architecture Neutral Distribution Format" or ANDF. ANDF is a proposed technology which will allow vendors "shrink-wrap" software which will run on a wide variety of computer systems independent of the CPU/Operating System combination. The only way to accomplish this effectively is to distribute the source code and allow it to be compiled on whatever system the user happens to have. Of course there are several obvious problems with this approach. It is much easier to pirate source code to a new system. The user can change the source code, recompile it, and end up introducing bugs in the program which the company will have no way of knowing what is going wrong. Also competitors can easily find out trade secret techniques used in a vendor's product.

Needless to say source code distribution is not practical for any software other than public domain software.

ANDF was proposed as a solution to this problem. ANDF represents a compromise approach somewhere between distributing a completely compiled binary and the original source code. With ANDF the software company partially compiles the source code into ANDF (this step is actually called "producing" rather than compiling in ANDF). ANDF is a representation of the source code which is easily converted into the final machine language for the target system. Each CPU/ operating system has a product called an "installer" which finishes the compilation process producing an executable for the system.

As I listened to the presentation, I became very excited about the possibilities for this new technology. Coupled with the possibility of OSF/1 becoming a ubiquitous operating system, ANDF promised a "brave new world" in which one could go to the local store, buy a single copy of a word processing package and run it on any system you can find.

What Happened?

We certainly cannot go out today and buy applications in ANDF and install them on our computers. Well, several things went wrong in what seemed to be a fairy tale with happy ending. First, ANDF cannot completely compensate for a lack of source code portability between systems. ANDF can solve many simple portability problems, like slight changes in parameter conventions, or the lack of a particular library routine on a particular system. But ANDF cannot convert an X-Window system application to a MS-Windows application. ANDF enhances existing portability between systems but does not create portability where no source code portability exists.

Also, vendors are a bit afraid to adopt ANDF. Portability is a two-way street. You can gain customers or you can lose customers. Certainly that appeals to users but it is very frightening to a vendor who has a tough enough time keeping customers from bolting. If all it takes to switch vendors is to buy a new system and "re-install" all of your software (assuming all the licenses were transferable) the buying public can become very fickle. Today, investments in software often are the key reason for staying with a particular vendors hardware.

The net result from my perspective (probably because I stopped going to OSF member meetings some time ago) is that ANDF fell off the face of the earth. I have not even heard the term mentioned by a vendor in the last few years.

As such, I was very much looking forward to the ANDF presentation at the recent POSIX meeting. For me it was like seeing an old friend at a 5-year class reunion.

So What's Up?

The ANDF BOF was well attended including presenters from OSF and the Defense Research Agency from the United Kingdom. About 20 people from POSIX attended the meeting. The first presentation was by Andrew Johnson of the OSF. The first part of his presentation was an overview of ANDF which summarized the goals and approaches that I described above. I was beginning to think that I was about to hear the same talk I heard before at OSF but there was more to come.

The ANDF format has been changed. The original ANDF was based on a compiler techniques from the 1970s. The format limited the potential for optimization for more advanced CPU architectures. The new ANDF format is a "tree-structured" representation of the program. This is the intermediate format used by most of today's highly optimizing compilers. This shift in intermediate format makes it possible to write an "installer" process which generates code that is as well optimized as any of today's vendors compilers.

OSF has compiled a number of demonstration applications. First, an ANDF version of the Spec92 benchmark executes within 5% of the performance obtained using the native compiler. And sometimes ANDF does better than the vendor compiler. They have also compiled several large production applications (1 million lines +) and have demonstrated that the ANDF versions execute with equivalent performance and reliability to the versions with native compilers.

In the next part of the presentation, they described a project which developed a front-end to gcc (public domain) which accepted ANDF as input and produced binaries using the rest of the `gcc` compiler. This can act as an installer function which can convert ANDF to any of the systems supported by the `gcc` compiler. They have put the source code to this ANDF installer in the public domain.

The second half of the presentation was given by the Defence Research Agency (DRA) of the UK. The DRA is one of the developers of the ANDF technology. DRA is working very hard on producing real products based on ANDF. They have spent a great deal of time on the reliability of the C producer. One of the challenges of a producer is to delay decisions typically done by a pre-compiler until just before the "link" phase of the compilation. This required the development of special versions of the C include files which captured the essence of function calls without actually generating function calls until the installer runs. This technology is called TICL (don't ask me what it stands for).

The TICL is generated directly from the text of a standard such as POSIX 1003.1. Because the TICL only allows the application to use features specifically described in the standard, it actually identifies areas where the application is using features beyond those specified in the standard. This aspect of TICL has some interesting uses in testing application conformance to standards.

Other organizations working with the DRA and OSF are expanding the suite of ANDF technology.

One of the goals of the BOF was to discuss the possibility of producing some sort of standard for ANDF. The ideas of making ANDF an X/Open standard and/or an IEEE standard were both discussed. While no specific plans were made, one or both of these organizations might be working on an ANDF standard within a year in my opinion.

There was a lot of discussion of how to move ANDF forward from the point of view of computer vendors, users, and application developers. Folks generally agreed that users had the most to gain followed by application developers. However users will not insist on ANDF if it is not available. This was termed the "chicken-egg-rooster" problem.

Of course there was a strong suggestion to put the entire ANDF kit into the public domain to encourage experimentation with the technology. OSF and DRA nodded wisely when this was brought up but did not have any solution in hand.

People interested in ANDF who haven't looked at it for a while should take another look. With nearly the entire kit in the public domain, there should be ample opportunity for folks to experiment. They have an ANDF binary of Mosaic available which would be an interesting test that we all could do.

For more information on ANDF, take a look at *http://riwww.osf.org:8001/r1/andf.papers/ANDF-intro.html* on the WEB and *riftp.osf.org* on anonymous FTP.

Report on NII/GII Information Infrastructure BOF

Charles Severance <crs@msu.edu> reports on the October 1994 meeting in Seattle, WA:

Last October, a Birds-of-a-Feather (BOF) session was held at the Seattle POSIX meeting to introduce the attendees to some of the issues in the National/Global Information Infrastructure (aka NII). The BOF was hosted by Jim Isaak, the acting chair of the IEEE SCC33, the IEEE standards coordinating committee that acts as a focal point for identifying areas for NII standardization for the IEEE.

In the first part of the presentation, Jim gave an overview of the organizations working on NII standardization.

- ANSI IISP – Information Infrastructure Standards Panel. This is a focal point for the Information Infrastructure standards requirements and responses.

- IEEE SCC33 – The focal point within the IEEE for Information Infrastructure standards efforts (Jim Isaak is the Acting Chair of this group).

- CSPP – Computer Systems Policy Project – This group consists of the CEOs of the major computer manufacturers. They have written a document which provides a high level vision of the NII.

- XIWT – Cross Industry Working Team – This group is made up of a broad range of computer, telecom, and cable companies (42).

In the second part of the presentation, the group discussed some of the broader issues in the NII. The first observation is that there are many perspectives on what the NII should actually become. Some folks believe that the Internet is the prototype for the NII. Others feel that the purpose is to have 500 cable channels with a little remote control so folks can make orders without using their telephones. Yet another perspective is that it is a super phone network with ISDN and beyond.

The good news is that because each of the industries represented (telephone, computer, and entertainment) is very powerful, the NII will most likely be a combination of the dream systems of all of the groups.

Report on POSIX.0: Guide to POSIX OSE

Kevin Lewis <klewis@gucci.enet.dec.com> reports on the January, 1995 meeting in Ft. Lauderdale, FL:

The best news a ballot coordinator can get is that the draft standard he is working on has completed recirculation with only 3 objections. Such was the case with the second recirculation for the POSIX.0 guide document. When the ballot closed December 19, a total of 6 ballots had been submitted containing 69 comments and 3 objections. Two of the 6 ballots were "NO" votes. One was converted, the other remains a negative vote. This brings the ballot results on the guide to the following figures:

81 eligible voters
56 affirmative votes
8 negative votes
64 total votes
= 87% AFFIRMATIVE

The pathway to approval is now clear. All comments and one of the three objections will be reflected in the next draft of the guide which will be draft 18. This will be submitted to the IEEE on February 3 for distribution to the IEEE Review Committee (RevCom). RevCom will meet on March 15 in Piscataway, NJ for final review at which time a recommendation to the IEEE Standards Board will be made. I anticipate no problems whatsoever in obtaining approval by the Board. I am told that once a draft is approved, it takes approximately 6 months for it to be published.

The March board meeting is quite symbolic in this development effort for it was exactly seven years ago at the March 1988 POSIX meeting that the working group had its first meeting. I've been writing as one of the original snitches since that time. I shudder to think how many times I forecasted (and in error, I might add) when this work would reach completion.

At the international level, draft 18 will be submitted to the SC22 Secretariat for the last ISO ballot which will be as a DTR (Draft Technical Report) within JTC1.

Now, as for the future, there are several directions that the POSIX.0 working group could go. Currently, POSIX.0 has spawned the OSE (Open Systems Environment) User Profile Study Group which is focusing on the development of a methodology for identifying OSE user requirements. In addition to this, several people have shown interest in starting work in the distributed systems area, commonly referred to as "middleware." Others have shown interest in examining how ODP and OSE should come together. And there are others who are interested in expanding the current guide document further, specifically in the reference model and services areas. Right now, the aforementioned study group is the only effort that has been organized. The other areas are waiting for a champion to come along. Could that be you?

Report on POSIX.6: Security Extensions

Lynne Ambuel <ambuel@dockmaster.mcsc.mil> reports on the January 1995 meeting in Fort Lauderdale, FL:

Security Ballot Reforms – and Lives to Tell about It

One curious thing about human nature is the compulsion to avoid change. The manager that fights every effort to automate the office. The worker in a dead-end job that doesn't accept the opportunity to move into a better one. The battered women who refuses to leave the abusive relationship. Stories upon stories are told of people who would just as soon stay in difficult, if not unbearable, situations because they know what to expect and they can't be sure that the change wouldn't be worse. It is funny how we can be so comfortable in our discomfort.

By 1994, much to my dismay, the Security Extensions ballot resolution group had fallen into this same paradigm. We had grown comfortable with our draft and the ballot group. The problem was we didn't realize how miserable we really were. By that time the ballot group was almost four years old. It had become an adventure to find the 275 people in the group every time a new draft was released for ballot. Once we found them it was even harder to make them care. It would take up to three months of concerted effort just to get enough ballots returned to close ballot. It was also a challenge to get enough non-abstentions to be a valid ballot in the eyes of IEEE. On several issues we had reached stalemate. (The ACL mask was swapped in and out at alternate drafts.) Looking back, we should have been screaming for a change.

But – nooooo. Early in 1993 the IEEE suggested that we re-form our ballot group. We resisted; quite adamantly, I might add. We knew what our balloters were going to say. A new group may have had a totally different idea of what the standard should contain. They wouldn't have the "history" and would set us further back by rehashing debates that were decided years ago. We knew our situation was bad, but we were sure that the change would be worse. By the end of 1993 IEEE insisted that we re-form our ballot group. We had no choice. There was great weeping and gnashing of teeth but we finally submitted.

Much to our surprise, things went smoothly. The new ballot group of 82 people was formed by the end of 1993. There were some snags in letting people know they needed to sign up, or re-sign up; but those were relatively mild. The new group was comprised up of a mixture of people from the old ballot group as well as new members (37%). Several people who had wanted to comment on the standard but had not been involved when the original ballot group was formed were able to join. Others that had lost interest and would either vote affirmative without reading the draft, abstain or not return the ballot at all did not rejoin. Somehow the new group contained a balance of background and experience that equaled, if not exceeded, that of the original group.

What was even nicer was the ballot closed within two days of the original closing date, without any prompting. The most difficult thing was warning the old-faithfuls that were used to having an extra month or so to finish their comments that the ballot period was about to close and they had better get their ballots to IEEE ASAP.

Even our fears that the comments and objections would send us in a new direction were ill-founded. Our affirmative votes dropped from 61% to 47%, however, very few ballots had any really hard issues. There were no new hard issues. There were some new points of view, but they mostly affirmed the general direction we had been taking. And yes, there were some objections that requested changes way out in left field. (Sorry we can't solve world hunger in the security standard.) This will take a little education time. All in all, the ballot resolution task set before us was the most manageable that it had been since we began the process of resolving ballots on this standard.

Because of all these factors we are about to release our next ballot draft (D15), a mere three meetings after closing the last one. We also believe that we have done enough work to significantly increase our affirmative votes. I think we will match our all time high of 61% and may even exceed it. We also think we can put out a follow-on draft by the end of the year, if the ballot is closed by the April POSIX meeting. That would be the first time we have released two drafts in the same year.

Change is good.

Dilbert ® by Scott Adams

DOGBERT, LOOK, I GOT THE FIRST VIDEO PHONE IN THE CITY!
S. Adams
NOW WE WAIT FOR SOMEBODY ELSE TO BUY A COMPATIBLE VIDEO PHONE AND CALL US.
© 1992 United Feature Syndicate, Inc
THE AMAZING THING IS THAT SOCIETY COULDN'T ADVANCE WITHOUT PEOPLE LIKE YOU.
I THINK I SAW SOME-THING.

DILBERT reprinted by permission of UFS, Inc.

The Bookworm

by Peter H. Salus
<peter@uunet.uu.net>

All my fault

I don't know exactly where the daemon crept into my last column. . . my machine, some other machine. . . but I want to begin this column with an apology to Mark Andreesen and his colleagues at the NCSC who created Mosaic. I did not mean to imply that both WWW and Mosaic were originated by Tim Berners-Lee at CERN. I also want to thank Guy Harris, who reads these columns carefully enough to catch my bloopers and point them out to me.

Servicing the Users

No matter what you read, there are many millions plying the Net these days. Perhaps only 10 million have a true Internet connection, perhaps more. But if you're going to provide a connection to the Internet from a UNIX (or UNIX-like) system, *Managing Internet Information Services* is the book you'll want. Peek, Liu and their colleagues have done a splendid job of putting together the data you'll need: two chapters on the internet and internet services; one on `finger`, `inetd`, and `telnet`; three on `ftp`; two on WAIS; eight on `gopher`; five on the Web; five on mailing lists (and `ftpmail`); and much, much more!

As a curmudgeonly historian, I'm sorry to say that while Lee McLoughlin gets credit for `ftpmail`'s PERL scripts, Paul Vixie gets zilch for having written and maintained `ftpmail` for years. (Of course, this isn't the original `ftp mail`, which Tomlinson wrote in 1970.)

This has nothing to do with installing and maintaining the varied services discussed. This is an ORA book in their brilliant tradition: thorough, hands-on, how-to.

Back to Basics

Kevin Reichard has turned out an interesting series of introductory books, under the general rubric of "UNIX Fundamentals" (actually, one is coauthored by David Burnette). I've gone through two of them and looked at a third. *UNIX for DOS and Windows Users* was opaque to me, as I've never used either DOS or Windows. *UNIX: The Basics* and *UNIX: Communications and Networking* are quite good. But all four books share a bizarre drawback: all the items in each of the four volumes are written by Reichard. I am as vain as any author, but I find it absurd and contemptible to pretend that the best or most informative volume on every topic is mine.

In *Shareware and Freeware*, Reichard states "The Internet was built with UNIX freeware." I am certain this sentence intended to mean something. Most of the volume reads like promo for the FSF and Linux.

Giving credit to authors just isn't where Reichard is at. You'll hunt in vain for Ritchie or Thompson or Johnson or . . . in *Basics*; while (Burnette's influence?) HoneyDanBer is mentioned in *Communications and Networking*, the fact that Mike Lesk originally wrote UUCP is not there. However, McGill and Minnesota are given credit for archie and gopher; Nevada and CERN for Veronica and WWW. While I think these volumes will be useful for beginners, I think that not giving credit to inventors and implementors is reprehensible.

Books reviewed in this column

Cricket Liu, Jerry Peek, Russ Jones, Bryan Buus, Adrian Nye, **Managing Internet Information Services**. O'Reilly & Associates, 1994. 630pp. ISBN 1-56592-062-7. $29.95

Kevin Reichard, **UNIX: The Basics**. MIS Press, 1994. 277pp. ISBN 1-55828-362-5. $19.95.

Kevin Reichard & David Burnette, **UNIX: Communications and Networking**. MIS Press, 1995. 246pp. ISBN 1-55828-388-9. $19.95.

Kevin Reichard, **UNIX: Shareware and Freeware**. MIS Press, 1995. 203pp. + CD-ROM. ISBN 1-55828-382-X. $24.95.

Kevin Reichard, **UNIX for Dos and Windows Users**. MIS Press, 1994. 283pp. ISBN 1-55828-361-7. $19.95.

Belinda Frazier, ed.,**The Linux Sampler**. SSC, 1995. 240pp. ISBN 0-916151-3. $18.95.

Harley Hahn and Rick Stout, **The Internet Yellow Pages**. 2nd ed., Osborne/McGraw-Hill, 1995. 812pp. ISBN 0-07-882098-7. $29.95.

Gerald L. Hopkins, **The ISDN Literacy Book**. Addison-Wesley, 1995. 367pp. ISBN 0-201-62979-8. $34.50.

Lyn Dupre, **Bugs in Writing**. Addison-Wesley, 1995. 649pp. ISBN 0-201-60019-6. $19.95.

Jerry Jongerius, **Writing Bug-Free C Code for Windows**. Prentice Hall, 1995. 218pp. ISBN 0-13-183898-9.

May R. Berenbaum, **Bugs in the System**. Addison-Wesley, 1994. 377pp. ISBN 0-201-62499-0. $25.

Bruce Schneier, **E-Mail Security**. Wiley, 1995. 352pp. ISBN 047105318X. $24.95.

Linux

Speaking of Linux, SSC, the folks in Seattle who publish *Linux Journal*, have put out a brief anthology of really interesting Linux-related material, *The Linux Sampler.* It's well-worth the investment both of dollars and time.

Yellow Pages

I have remarked several times on the fact that there may well be too many Internet books now. The second edition of Hahn and Stout's *Internet Yellow Pages* isn't one of the otiose (*Webster says it means "functionless." Ed.*) ones. I put it next to my box for several weeks. I looked for stuff in it. I found the entries; and I found the stuff. The value of a reference book is in its utility; this one works for me.

ISDN

There are two different views of ISDN: It Still Does Nothing vs. Information Superhighway Delivered Now. Bob Metcalfe is biased towards the latter case, I admit to leaning towards the former. But Hopkins' book makes it far easier for me to understand just what some folks are enthusiastic about. In case you've been hiding on a speck in Micronesia without a phone, Integrated Services Digital Network is a telecommunications network that purports to be able to move text, music, video, images, fax, etc., across a single access line. According to Hopkins, this is the access medium of the future. It may well be. But now I think I understand many of the details, including how it will (would?) affect the end-users.

Entomological Notes

The original computer bug is in the Smithsonian; a picture of it was in *Annals of the History of Computing* vol. 3 (1981). The photos are of the log pages of the Naval Surface Warfare Center for September 9, 1947, and include the moth that had been trapped between the relay contacts. Of course, as moths are lepidoptera, not hemiptera, they aren't *true* bugs. The word "bug" for "a problem" antedates this use by a good deal, however; it goes back at least to Edison's time a century ago.

I got three books with "bug" in their titles recently. *Bugs in Writing* is cute-sy, but not something I would recommend to anyone interested in improving their writing. *Writing Bug-Free C Code for Windows* suffers from the ministrations of marketeers: if you get as far as the Preface, you will discover that the author thinks that with his methodology "you will produce code that contains fewer bugs." I think that this is probably valid, as the volume is really about a class methodology that eliminates data structures from include files. It uses private class declarations, rather than the public class declarations of C++. I'm not sure a whole book was needed.

The third book was *Bugs in the System.* It's an entomology book. It's terrific! Based on the author's class "Insects and People," offered at the University of Illinois, this book is as good as a novel. The author is full of anecdotes and possesses a weird sense of humor. As the punched cards of the Jacquard loom were for weaving silk, there is even a passing reference to Hollerith and IBM. More seriously, this is a wonderfully entertaining piece of work and recommended most highly.

I think that Schneier's *E-mail Security* deserves a mention. This is a solid, well-written book on PGP and PGM. It is not as stunning as Schneier's *Applied Cryptography*, but it is a good book. I think I prefer Simson Garfinkel's PGP book, reviewed last month, but this is no slouch.

Advertisement

By the time you read this, my *Casting the Net: From ARPANET to Internet and Beyond* should be out (Addison-Wesley). It is full of real information, lots of maps, many interviews, and a terrific Foreword by Vint Cerf.

Exploring Expect: A Tcl-based Toolkit for Automating Interactive Programs

by Don Libes (O'Reilly & Associates, Inc., 1995, ISBN: 56592-090-2.) 568pp.

Reviewed by Glenn Vanderburg <glv@utdallas.edu>

I can recall my reaction when I first saw an announcement for Don Libes' *Exploring Expect*: "550+ pages about *expect*? You've got to be kidding!" It's now a couple of months later, I've read all of those pages, and I'm no longer so incredulous. There's more to `expect` – or, more accurately, to its application domain – than I had realized.

`Expect` was designed to automate and control interactive programs, which are often very perverse. Libes explains `expect` and those applications side by side, demonstrating techniques for handling strange situations through a carefully planned series of examples. Along the way, the reader comes to understand the reasons behind some of `expect`'s features, and gains a feel for the many (often surprising) ways that `expect` can be put to use.

In addition, there's more here than just `expect` proper. `Expect` is a Tcl application, and effective use of `expect` requires a knowledge of Tcl, so a nice Tcl overview is included. `Expect` comes with a Tcl debugger, which is explained. There are sections on using expect with Tk (to provide X interfaces to character-based applications), using expect from C or C++ instead of Tcl, and using expect with other Tcl extensions.

It's clear that a lot of work went into this book. The examples are real, useful programs, not contrived exercises, and they are presented and explained well. Furthermore, I got the impression that Libes kept track of questions users asked him, and drew on them to make sure that the book covers all of the tricky areas. Exercises at the end of each chapter are thought-provoking and serve to emphasize the points of the chapter. And the index is very thorough. That, plus two mini-indexes, one to expect commands, options, and variables, and the other to examples, make this tutorial-style book a passable reference. I would have liked it if the manual page had been included, but it isn't a glaring omission.

All in all, *Exploring Expect* is a fine book about a fine and versatile tool.

UNIX Systems for Modern Architectures, Symmetric Multiprocessing, and Caching for Kernel Programmers

By Curt Schimmel (Addison-Wesley, 1994, ISBN: 0-201-63338-8.) 396pp.

Reviewed by George V. Neville-Neil <gnn@netcom.com>

UNIX has come a long way from its beginnings on DEC's PDP computers in the early 1970s. Now, more than 25 years since UNIX was first developed, it has been ported to many different vendors' hardware. In the last decade, new systems/processors have been developed which employ several mechanisms that were not envisioned by the original UNIX designers.

One performance enhancement that has been added to microprocessors is a cache for holding information that is frequently referenced. Caches exploit locality of reference; the idea is that programs frequently reference the same data in a small period of time. The cache holds a copy of data from main memory in a faster memory closer to (or actually on) the CPU. This can shorten the time it takes for the CPU to access a piece of information since it does not have to go across the memory bus.

Many systems are also built with more than one CPU. These multiprocessor, or MP, systems try to improve overall system performance by using many cheaper processors instead of a single, faster, and possibly more expensive processor.

This book discusses these subjects as they affect a kernel programmer implementing/porting a traditional single-CPU UNIX system on hardware that has a cache, multiple processors, or a combination of the two. Contemporary hardware from a variety of vendors (including MIPS, TI, Motorola, and Intel) is used for the examples presented throughout.

After briefly reviewing the UNIX Kernel Internals that are pertinent to these types of systems in Chapter 1, the book begins with "Part I: Cache Memory Systems." Each chapter in this section presents a different type of cache. The types are differentiated by the way in which they see memory. The chapters are further broken down by how the cache must be managed during: context switch, fork, exec, exit, brk and sbrk, shared memory and mapped files, I/O, and user-kernel data ambiguities. The final chapter in this section discusses some design trade offs that can be made to make more efficient use of a cache.

Part II presents "Multiprocessor Systems." The first chapter is an introduction to MP systems, and lays out the MP hardware that will be used in the following chapters. The next chapters discuss the different ways in which a single processor operating system (such as UNIX) can be made to run on an MP system while retaining its programming model. Most of this has to do with allowing the kernel to run on more than one processor while keeping the kernel's data-structures from being corrupted. Each chapter introduces new O/S primitives that allow the kernel's work to be distributed to more and more processors while preserving its data integrity.

"Part III: Multiprocessor Systems with Caches" combines the mechanisms described in the first two parts. There are only two chapters in this section. They both concentrate on the problems of maintaining cache consistency when there may be more than one processor accessing the same piece of main memory but each caching its own copy of the data. This problem, cache consistency, is solved in hardware in the last chapter.

Each chapter concludes with a set of exercises for the reader, as well as a glossary for further reading. The answers to selected exercises are presented in Appendix B. Appendix A is a summary of some current microprocessors that covers: Apollo DN 4000, Intel (80386, 80486, 82495 DX, Pentium, i860 XR, i860 XP), MIPS (R2000/R3000 & R4000), Motorola (68040 & 88000), Sun 3/200 and TI (MicroSPARC & SuperSPARC).

The whole book is written in an easy-to-read style that I enjoyed. Part I, on caching, is an excellent introduction to how caching works, with good, clear examples, using modern hardware. I felt that Part II was somewhat unfocused.

Though the book is targeted at kernel programmers, Part II spends a great deal of time describing semaphores, spin locks, and other primitives that should be familiar to any operating system programmer. Part III seemed even less focused as most of the problems it presented need never bother someone programming at the kernel level; they are solved in hardware.

This book is a worthwhile introduction to the subjects of caching and symmetric multiprocessing but it is not exhaustive. If you have little or no knowledge of these areas and wish to learn about them then you should pick up this book. If, on the other hand, you have even a passing knowledge of these subjects, this book is not for you.

O'Reilly's Internet Talk Radio Tapes

by Rik Farrow
<rik@spirit.com>

I don't know about you, but I can't listen to a talking tape and get any work done. O'Reilly and Associates, by sending me their complete set of *Internet Talk Radio* tapes, had, implicitly, attempted to convince me that I should at least attempt this. But the tapes languished on my book shelves until I found myself traveling more, spending long hours commuting across the Arizona desert to the airport. Suddenly, the notion of talking tapes, with intelligent people speaking, became much more appealing.

Carl Malamud hosts *Internet Talk Radio*, which means that you can get the contents of most of these tapes via ftp, and play them as audio files. But my car doesn't handle audio files as well as it does stereo cassettes, but O'Reilly's tapes make sense for anyone with a long commute who is bored of listening to disk jockeys.

My favorite tapes were about Internet topics, such as the *Future of the Interne*t, or *Managing the Internet.* Sound quality is generally good, and I found I learned lots of things listening to interviews with Mike O'Dell or Phil Cairns. Even the tape of EFF's spokesperson turned out to be great (I really missed a lot when I heard his USENIX keynote the first time).

I haven't seen any new tapes from O'Reilly recently, and find I wanted to hear more as I cruise the desert. Sure beats AM radio.

About Refdbms and USENIX References

by Richard Golding
<golding@cello.hpl.hp.com>

Refdbms is a distributed bibliographic system that allows different Internet sites to maintain replicas of databases of references to articles, technical reports, and so on. At present, 56 databases containing more than 29,000 references are available; many of the references include abstracts and location information. We presented a paper on the system at the 1994 Winter conference in San Francisco, but you can get details (and try the system out) using my web home page: *http://pegasus.esprit.ec.org/people/golding/index.html.*

I (try to!) keep the USENIX Online Library up-to-date by periodically updating from the file maintained by the USENIX office and by entering references from those conferences I attend. To enter the OSDI and New Orleans conference references, I scanned the first page of each paper to get a text file containing titles, names, affiliations, and abstracts. I hand-inserted these data into a boilerplate template for each reference, spell- and syntax-checked the lot, and was done. In total it took just over an hour for the Winter conference references, and was much easier on the hands than typing the lot from scratch.

Announcement and Preliminary Call for Papers

USENIX Workshop on Electronic Commerce

Electronic Commerce Workshop
July 11-12, 1995
Electronic Commerce Seminars
July 13-14, 1995
Sheraton New York Hotel & Towers
New York, New York

The First USENIX Workshop on Electronic Commerce will provide the major opportunity for researchers, experimenters, and proto-practitioners in this rapidly self-defining field to exchange ideas and present results of their work. This meeting will set the technical agenda for work in the area of Electronic Commerce by deriving and/or certifying the most urgent questions, discovering directions in which answers might be pursued, and revealing cross-connections that otherwise might go unnoticed. Marketing hysteria this is not; technical direction setting it is.

Format

This four-day event has an unusual structure; its first two days will follow a workshop format at which attendance is by invitation, while the latter two days will be tutorial in nature and open to all. The multi-tracked workshop will feature refereed and position paper presentations, reports of works-in-progress, technological debates, and identification of hard-to-impossible problems. Birds-of-a-Feather sessions, a dinner speaker, and a Keynote speaker will round out these two very full days and nights.

The following two days, July 13-14, will offer four half-day tutorials bracketing an evening session of free-form Q&A with technical experts, called a "Guru is In" at other USENIX meetings. The half-day tutorials will be repeated on a rolling basis, so that participants will be able to attend most of them. Acknowledged leaders in this critical, cutting-edge technology will lead the tutorials and serve as sounding boards during this portion of the Workshop.

Topics

The Workshop on Electronic Commerce will address a wide range of issues and ongoing developments, including, but not limited to:

Technical security

Mutual authenticity
Integrity
Confidentiality
Non-repudiability
Cryptographic mix and match
APIs of appropriate abstraction level

Business security

Insurance
Bonding
Liability
Prompt revocation guarantees
Counterparty identification
Credit history
Integrity requirements for long term record retention

Finance and Payment

Digital cash and credit
Trans-border clearance commerce
Direct, broker-less markets
Auditability versus privacy
EDI and its role, ditto X9.17

The Commons

Address management problems, IP addresses going fast
Threats to 'flatness' — hidden spaces and toll roads

Legal matters

Border-less-ness of an electronic marketplace
Crimes of the future

Potential mass audiences

Email-enabled business
Internet service for technically unsophisticated clients
Management by subscription and other invasive conveniences

Service level guarantees in Internet service settings

Latency
Privacy
Outages/downtime
Bandwidth to specific customers
Cross traffic and guarantees thereon
Mobility/intermittency
Time synchronization and absolute ordering

Advertisement and Service Access

Discovery: White Pages/ Yellow Pages
Competitive issues
Interaction with intelligent agent search
Active advertisements and payment mechanisms

Services and Their Components

What is salable: if content becomes free, indexing becomes dear
Franchising and delegation
Reliability
Transactional services

Participation in the Workshop (July 11-12)

In a departure from usual USENIX events, attendance will be by invitation and we will cap attendance at 100. All persons wishing to attend are requested to submit a formal paper (via extended abstract) for a conventional referee process or, where less formal submission is desirable, a position paper, a statement of intent, a description of work in progress, or, some other communication of the nature of their contribution.

We seek original and innovative papers, demonstrations, videotapes, position papers and general smart work about current developments in electronic commerce, its precursors and/or its infrastructure. We are especially interested in reports on practical experiences with such systems, if indeed there yet be any. Formally refereed papers will be published in the Proceedings, distributed free to attendees of the Electronic Commerce Workshop and USENIX members, and later made available for purchase from the USENIX Association. Non-refereed papers, broadsides, commercial siren songs and more, all likely to be in evidence, are appropriate and encouraged, and will be praised or debunked as they warrant.

Submission Guidelines

(1) Refereed papers — For those with formal papers for inclusion in the Proceedings, the referee process will take place in a conventional manner:

Submission of an extended abstract of 1500-2500 words (9000- 15000 bytes or 3-5 pages) is recommended. Shorter abstracts run a greater risk of rejection as there will be little on which the program committee can base an opinion.

For administrative reasons, please include a separate page or (preferably) e-mail message giving the title of the paper, the names and affiliations of the authors, and the name of the author who will act as the single point of contact for the Organizing Committee. For the contact person, also include a daytime telephone number, postal address, email address, and fax number if possible.

If you would like to receive detailed guidelines for submission and past examples of accepted extended abstracts, you may telephone the USENIX Association office at 510 528 8649, or email to *ec95authors@usenix.org.*

Dates for refereed paper submissions

Extended abstracts due: *May 8, 1995*
Notification to authors: *May 22, 1995*
Camera-ready final papers due: *June 15, 1995*

(2) Non-Refereed submissions — All persons who do not have available a reviewable paper must submit a position paper, a statement of intent and direction, a short description of work in progress to varying degrees that can be discussed with the other attendees, or, in some other way, demonstrate your resolve to contribute substantially and insightfully to this Workshop. There are no hard and fast rules here and we by no means wish to discourage any tentative interest, but the Organizing Committee anticipates that the program will fill quickly and encourages you to anticipate our need to make the meeting itself as successful as possible. This is a first meeting in this topic area, it will almost surely not be the last, and we intend that it set the agenda for further work and progress.

USENIX symposia, like most symposia and journals, require that papers not be submitted simultaneously to more than one conference or publication and that submitted papers not be previously or subsequently published elsewhere. Submissions accompanied by "non-disclosure agreement" forms are not acceptable and will be returned to the author(s) unread. All submissions are held in the highest confidentiality prior to publication in the Proceedings, both as a matter of policy and in accord with the U.S. Copyright Act of 1976.

For questions about submissions and other program concerns, contact the Program Chair, Dan Geer.

Submissions should be sent to:
ec95papers@usenix.org
in electronic form. Failing that, send them to the program chair,

Daniel E. Geer, Jr., Sc.D.
OpenVision Technologies, Inc.
One Main Street
Cambridge, MA 02142
617 374 3700

Organizing Committee

Chair:

Daniel E. Geer, Jr., Sc.D.
OpenVision Technologies, Inc.
geer@cam.ov.com

Joseph Arceneaux
Wells Fargo Bank
joseph.arceneaux@wellsfargo.com

Nathaniel Borenstein
First Virtual Holdings, Inc.
nsb@fv.com

Marc D. Donner
Morgan Stanley & Co., Inc.
donner@fid.morgan.com

Steve Dusse
RSA Data Security, Inc.
spock@rsa.com

Ittai Hershman
Advanced Networks & Service Inc.
ittai@nis.ans.net

Peter Honeyman
CITI, University of Michigan
honey@citi.umich.edu

Alan Nemeth
Digital Equipment Corporation
agn@zk3.dec.com

Cliff Neuman
Information Sciences Institute
bcn@isi.edu

Greg Rose
Sterling Software Pdy. Ltd.
ggr@sydney.sterling.com

Allan M. Schiffman
Enterprise Integration Technologies Corp.
ams@eit.com

Mark Seiden
Seiden and Associates, Inc.
mis@seiden.com

Win Treese
Open Market, Inc.
treese@OpenMarket.com

Others TBD shortly

Registration Information

Materials containing all details of the tutorial programs, registration fees and forms, and hotel information will be available in mid-April, 1995. If you wish to receive the registration materials, please contact USENIX at:

USENIX Conference Office
22672 Lambert Street, Suite 613
Lake Forest, Calif. 92630
714 588 8649, fax 714 588 9706
conference@usenix.org

For more information about USENIX and its events, access the USENIX Resource Center on the World Wide Web. The URL is *http://www.usenix.org.* OR send email to our mailserver at *info@usenix.org.* Your message should contain the line: *send catalog.* A catalog will be returned to you.

January 22-26, 1996
San Diego Marriott Hotel and Marina
San Diego, California

USENIX

technical conference 1996

Announcement and Preliminary Call for Papers

Dates for Refereed Paper Submissions

Extended Abstracts or Manuscripts Due: *July 18, 1995*
Notification to Authors: *August 31, 1995*
Camera-ready Papers Due: *November 14, 1995*

Conference Schedule Overview

Tutorials: *January 22-23, 1996*
Technical Sessions: *January 24-26, 1996*
Birds-of-a-Feather Sessions: *January 23-25, 1996*
USENIX Reception: *January 24, 1996*
Vendor Display: *January 24-25, 1996*

Program Committee

Robert Gray, *US WEST, Program Chair*
Miche Baker-Harvey, *Digital Equipment Corporation*
David Black, *Open Software Foundation*
Matt Blaze, *AT&T Bell Laboratories*
Darrell Long, *University of California at Santa Cruz*
John Ousterhout, *Sun Microsystems*
Christopher Small, *Harvard University*
Jonathan Smith, *University of Pennsylvania*
Carl Staelin, *Hewlett-Packard*
Brent Welch, *Xerox PARC*

Call for Submissions

The 1996 USENIX Technical Conference Program Committee seeks original and innovative papers about the applications, architecture, implementation, and performance of modern computing systems. As at all USENIX conferences, papers that analyze problem areas and draw important conclusions from practical experience are especially welcome. Some particularly interesting hot application topics are:

Privacy and cryptography
Compression applications
User interface toolkits
Nomadic and wireless computing
Networks and distributed computing
Security

A major focus of this conference is operating systems practice and experience. We seek papers describing original work and results in the design, implementation, and use of modern operating systems. Submissions describing extensions or modifications of complete and widely used operating systems are particularly encouraged in addition to those describing research systems or prototypes. Topics of interest in this area include but are not limited to:

OS structure and organization
Performance and optimization
OS support for real time & multimedia
OS support for embedded systems
OS interaction with HW architecture
Microkernel internals, servers, and applications

Note that the USENIX conference, like most conferences and journals, requires that papers not be submitted simultaneously to more than one conference or publication, and that submitted papers not be previously or subsequently published elsewhere. Papers accompanied by non-disclosure agreement forms are not acceptable and will be returned to the author(s) unread. All submissions are held in the highest confidentiality prior to publication in the Proceedings.

Cash Prizes

Cash prizes will be awarded for the best paper at the conference and the best paper by a full-time student.

How to Submit a Refereed Paper

It is important that you contact the USENIX Association office to receive detailed guidelines for submitting a paper to the refereed track of the technical sessions. Please telephone 510 528 8649 or send email to: *usenix96authors@usenix.org.*

In addition, specific questions about submissions to the USENIX 1996 Technical Conference may be made to the program chair via email at: *gray@usenix.org.*

The program committee will review full papers or extended abstracts. An extended abstract should be five manuscript pages (single-sided) or fewer in length. It should represent the paper in "short form." If the full paper has been completed, it may be submitted instead of an extended abstract. Full papers should be limited to 12 single-spaced pages.

Include references to establish that you are familiar with related work, and, where possible, provide detailed performance data to establish that you have a working implementation and measurement tools.

Where to Send Submissions

Please send one copy of an extended abstract to the program chair via one of the following methods. All submissions will be acknowledged.

Preferred Method: email (PostScript or ASCII) to *usenix96papers@usenix.org*

Alternate Method: postal delivery to
Robert Gray
U S WEST Technologies
4001 Discovery Drive, Suite 280
Boulder, CO 80303
Phone: 303 541 6014

USENIX®, the Advanced Computing Systems Professional and Technical Association.

The authors must also submit via email to *usenix96papers@usenix.org* the following information:

1. The title of the manuscript and the names of the authors.
2. The name of one author who will serve as a contact, an email address, day and evening phone numbers, postal mail address, and a fax number, if available.
3. An indication of which, if any, of the authors are full-time students.
4. A short abstract of the paper (75-150 words).

Tutorials

On Monday and Tuesday, you may attend intensive, immediately practical tutorials on topics essential to the use, development, and administration of UNIX and UNIX-like operating systems, windowing systems, networks, advanced programming languages, and related technologies. The USENIX Association's well-respected program offers you both introductory and advanced courses, presented by skilled teachers who are hands-on experts in their topic areas.

USENIX will offer two days of tutorials covering topics such as:

System administration
Systems and network security
Distributed computing: DCE, DFS, RPC, CORBA
Kernel internals: SVR4, Chorus, Windows NT
Systems programming tools and program development
Portability and interoperability
Client-server application design and development
Sendmail, DNS, and other networking issues
GUI technologies and builders

To provide the best possible tutorial slate, USENIX constantly solicits proposals for new tutorials. If you are interested in presenting a tutorial, contact the Tutorial Coordinator:

Daniel V. Klein
Phone: 412 421 2332
Email: *dvk@usenix.org*

Invited Talks

An Invited Talks track complements the Refereed Paper track. These talks by invited experts provide introductory and advanced information about a variety of interesting topics, such as using standard UNIX tools, tackling system administration difficulties, or employing specialized applications. Submitted Notes from the Invited Talks are published and distributed free to conference technical sessions attendees. This track also includes panel presentations and selections from the best presentations offered at 1995 USENIX conferences and symposia.

The Invited Talks coordinators welcome suggestions for topics and request proposals for particular Talks. In your proposal, state the main focus, include a brief outline, and be sure to emphasize why your topic is of general interest to our community. Please submit via email to: *ITusenix@usenix.org.*

Invited Talks Coordinators
Mary Baker, *Stanford University*
Ed Gould, *Digital Equipment Corporation*

Work-in-Progress Reports

Do you have interesting work you would like to share, or a cool idea that is not ready to be published? Work-in-Progress Reports are for you! Work-in-Progress Reports, scheduled during the technical sessions, introduce interesting new or ongoing work. The USENIX audience provides valuable discussion and feedback. We are particularly interested in presentation of student work. To schedule your report, contact Peg Schafer via email at *wips@usenix.org.*

Birds-of-a-Feather Sessions

The always popular Birds-of-a-Feather sessions (BOFs) are very informal gatherings of persons interested in a particular topic. BOFs often feature presentations or demonstrations followed by discussion, announcements, and the sharing of strategies. BOFs are offered Tuesday, Wednesday, and Thursday evenings of the conference. BOFs may be scheduled at the conference or in advance by telephoning the USENIX Conference office at 714 588 8649 or via email to: *conference@usenix.org.*

Vendor Display

In the USENIX Vendor Display, emphasis is on serious answers from technically savvy vendor representatives. Vendors will demonstrate the technical innovations which distinguish their products. Here, in a relaxed environment, conference attendees can ask questions and discuss how something will work with what they already have. Plus, you can review the newest releases from technical publishers.

Vendors: This is an exceptional opportunity for receiving feedback from USENIX's technically astute conference attendees. If your company would like to display its products and services, please contact: Zanna Knight. Telephone: 510 528 8649 Fax: 510 548 5738 Email: *display@usenix.org*

Conference Program and Registration Information

Materials containing all details of the technical sessions and tutorial program, conference registration, hotel and airfare discounts, and reservation information will be available at the end of September 1995. If you wish to receive the registration materials, please contact:

USENIX Conference Office
22672 Lambert St., Suite 613
Lake Forest, CA USA 92630
Phone: 714 588 8649
Fax: 714 588 9706
Email: *conference@usenix.org*
WWW URL: *http://www.usenix.org.*

Or send email to our mailserver at: *info@usenix.org.* Your message should contain the line: *send catalog.* A catalog will be returned to you.

The USENIX Association

Since 1975 the USENIX Association has brought together the community of engineers, scientists, and technicians working on the cutting edge of the computing world. The USENIX technical conferences and symposia have become the essential meeting grounds for the presentation and discussion of the most advanced information on new developments in all aspects of computing systems.

The USENIX Association and its members are dedicated to:

- problem-solving with a practical bias,
- fostering innovation and research that works,
- communicating rapidly the results of both research and innovation, and
- providing a neutral forum for the exercise of critical thought and the airing of technical issues.

SAGE, the System Administrators Guild, a Special Technical Group within the USENIX Association, is dedicated to the recognition and advancement of system administration as a profession.

2ND USENIX Mobile and Location-Independent Computing Symposium

April 10-11, 1995 Ann Arbor, Michigan

Preliminary Program Monday, April 10, 1995

9:00AM Opening Remarks:

Jim Rees, University of Michigan

9:15-10:30 Keynote Address:

Barry M. Leiner, Senior Scientist, Universities Space Research Association and currently on loan to the Advanced Research Projects Agency, where he is Deputy Director of the Computing Systems Technology Office.

11:00-12:30 CONNECTIVITY

Session Chair: Phil Karn

Routing in Mobile Wireless Networks

P. Krishna, M. Chatterjee, N.H. Vaidya, D.K. Pradhan, Texas A&M University

Handoff and Systems Support for Indirect TCP/IP

Ajay Bakre, B.R. Badrinath, Rutgers University

A Wireless Adapter Architecture for Mobile Computing

John Trotter, Mark Cravatts, AT&T Bell Laboratories

2:00PM-3:30 SIMULATION, EMULATION, AND ADAPTATION

Session Chair: Dan Geer

MCE: An Integrated Mobile Computing Environment and Simulation Testbed

Ramki Rajagopalan, Sridhar Alagar, S. Venkatesan, University of Texas, Dallas

A Network Emulator to Support the Development of Adaptive Applications

Nigel Davies, Gordon S. Blair, Keith Cheverst, Adrian Friday, University of Lancaster

A Programming Interface for Application-Aware Adaptation in Mobile Computing

Brian Noble, M. Satyanarayanan, Morgan Price, Carnegie Mellon University

4:00-5:00 PANEL DISCUSSION – Mobile Routing and Networks

Panelists: David Johnson, Carnegie Mellon University; Ramon Caceres, AT&T Bell Laboratories; Ajay Bakre, Rutgers University; Raj Yavatkar, University of Kentucky

6:00-8:00 Reception at the Ann Arbor Hands-On Museum

Tuesday, April 11, 1995

9:00AM-10:30 Work-in-Progress Reports

11:00-12:30 DISCONNECTIVITY

Session Chair: Dan Duchamp

System Isolation and Network Fast Fail Capability in Solaris

Gabriel Montenegro, Steve Drach, Sun Microsystems Inc.

A Generic Multicast Transport Service to Support Disconnected Operation

Silvano Maffeis, University of Zurich and Cornell University; Walter Bischofberger and Kai-Uwe Maetzel, Union Bank of Switzerland

Partially Connected Operation

L.B. Huston, P. Honeyman, CITI, University of Michigan

2:00PM-3:30 ENERGY AND MOBILITY

Session Chair: Jim Kempf

A Distributed Software Architecture for GPS-Driven Mobile Applications

Thomas G. Dennehy, Environmental Research Institute of Michigan

Energy Efficient Data Filtering and Communication

Tomasz Imielinski, Monish Gupta, Sarma Peyyeti, Rutgers University

Adaptive Disk Spin-down Policies for Mobile Computers

Fred Douglis, AT&T Bell Laboratories; P. Krishnan, Brown University; Brian Bershad, University of Washington

4:00-5:30 PANEL DISCUSSION – What do Applications Need to Know About Mobility?

Panelists: To be announced

Program Committee

Jim Rees, Program Chair, University of Michigan; Dan Duchamp, Columbia University; Dan Geer, OpenVision Technologies; Phil Karn, Qualcomm; Jim Kempf, Sun Microsystems; Inc. Jay Kistler, Digital Equipment Corporation

LISA'95

SEPTEMBER 18–22, 1995

SYSTEMS ADMINISTRATION CONFERENCE (LISA'95)

SEPTEMBER 18–22, 1995
MARRIOTT HOTEL
MONTEREY, CALIFORNIA

IMPORTANT DATES

REFEREED PAPER SUBMISSIONS:

- **Extended abstracts due: May 1, 1995**
- **Notification to authors: June 5, 1995**
- **Camera-ready papers due: August 1, 1995**

REGISTRATION MATERIALS AVAILABLE:

- **July, 1995**

LISA'95 WORKSHOP: ADVANCED TOPICS IN SYSTEM ADMINISTRATION WORKSHOP

- **Proposals due: August 1, 1995**
- **Notification to authors: August 14, 1995**

A one-day workshop, to be held Tuesday, September 19, 1995, will focus on a discussion of the latest-breaking technical issues in the systems administration arena as introduced by those in attendance. Attendance is limited and based on acceptance of a position paper. Acceptance notices to all participants will be issued by August 14, 1995.

HOW TO SUBMIT:

Potential workshop attendees are invited to submit a proposal of at most 3 pages (ASCII) via electronic mail to *jes@sgi.com* no later than August 1. These proposals should contain a topic for discussion, a description of the subject, an explanation of what makes this topic controversial or interesting, and a personal position. (More substantive reports of completed works should be submitted as papers to the technical sessions instead.) A representative subset of positions will be discussed in an open forum.

John Schimmel of Silicon Graphics is organizing this workshop. Email your proposal to *jes@sgi.com* by August 1. Chosen participants will be notified by August 14. Participants must be pre-registered for this LISA conference. No additional fee will be charged to attend this workshop, and lunch will be provided.

ANNOUNCEMENT & CALL FOR PARTICIPATION

The Systems Administration (LISA '95) Conference, sponsored by USENIX and SAGE, is widely recognized as the leading technical conference for system administrators. Historically, LISA stood for "Large Installation Systems Administration," back in the days when having a large installation meant having over 100 users, over 100 systems, or over one gigabyte of disk storage. Today, the scope of the LISA conference includes topics of interest to system administrators from sites of all sizes and kinds. What the conference attendees have in common is an interest in solving problems that can-not be dealt with simply by scaling up well-understood solutions appropriate to a single machine or a small number of workstations on a LAN.

The theme for this year's conference is "New Challenges," which includes such emerging issues as integration of non-UNIX and proprietary systems and networking technologies, distributed information services, network voice and video teleconferencing, and managing very complex networks. We are particularly interested in technical papers that reflect hands-on experience, describe fully implemented and freely distributable solutions, and advance the state of the art of system administration as an engineering discipline.

TUTORIAL PROGRAM

Monday and Tuesday, September 18–19, 1995

The two-day tutorial program offers up to five tracks of full and half-day tutorials. Tutorials offer expert instruction in areas of interest to system administrators of all levels, from novice through senior. Topics are expected to include networking, advanced system administration tools, Solaris and BSD administration, Perl programming, firewalls, NIS, DNS, Sendmail, and more.

To provide the best possible tutorial offerings, USENIX continually solicits proposals for new tutorials. If you are interested in presenting a tutorial at this or other USENIX conferences, please contact the tutorial coordinator, Daniel V. Klein: Phone: 412 421 0285; Fax: 412 421 2332; Email: *dvk@usenix.org*

TECHNICAL SESSIONS

Wednesday through Friday, September 20–22, 1995

The three days of technical sessions consist of two parallel tracks. The first track is dedicated to presentations of refereed technical papers. The second track will accommodate invited talks, panels and Works-in-Progress (WIP) sessions.

CONFERENCE TOPICS

Papers addressing the following topics are particularly timely; papers addressing other technical areas of general interest are equally welcome.

- Your plans for the year 2000
- Deployment of new networking technologies
- Coping with the commercialization of the Internet
- Support models in use at your site
- Dealing with differences in UNIX implementations – migration and interoperability among BSD, SVR4, OSF and others
- Integration of UNIX-based with non-UNIX-based and proprietary systems and networking technologies (Mac, NT and DOS PCs)
- Application of emerging technologies (Mbone, Mosaic) to system administration
- Administration and security of distributed information services (WAIS, gopher, WWW) and network voice and video teleconferencing (Mbone)
- Experience supporting mobile and location-independent computing
- Experience with large (1000+ machine) networks, especially networks of SVR4-based systems
- Real-world experience with implementations of proposed system administration standards
- Unusual applications of commercial system administration software packages
- Application of operational planning techniques to system administration including measurements and metrics, continuous process improvement, automation, and increasing productivity
- File migration, archival storage & backup systems in extremely large environments

2/20/95

- Innovative tools and techniques that have worked for you
- Managing high-demand and high-availability environments
- Migrating to new hardware and software technologies
- Administration of remote sites that have no technical experts
- Supporting MIS organizations on UNIX
- Real-world experiences with emerging procedural/ethical issues – e.g., developing site policies, tracking abusers, and implementing solutions to security problems
- Networking non-traditional sites (libraries, museums, K–12)

REFEREED PAPER SUBMISSIONS

An extended abstract is required for the paper selection process. Full papers are not acceptable at this stage; if you send a full paper, you must also include an extended abstract. "Extended" means 2–5 pages. Include references to establish that you are familiar with related work, and, where possible, provide detailed performance data to establish that you have a working implementation or measurement tool.

Submissions will be judged on the quality of the written submission, and whether or not the work advances the state of the art of system administration. For more detailed author instructions and a sample extended abstract, send email to: ***lisa9authors@usenix.org*** or call the USENIX office at 510 528 8649.

Note that LISA '95, like most conferences and journals, requires that papers not be submitted simultaneously to more than one conference or publication and that submitted papers not be previously or subsequently published elsewhere. Papers accompanied by "non-disclosure agreement" forms are not acceptable and will be returned unread. All submissions are held in the highest confidence prior to publication in the conference proceedings, both as a matter of policy and as protected by the U.S. Copyright Act of 1976.

Authors of an accepted paper must provide a final paper for publication in the conference proceedings. At least one author of each accepted paper presents the paper at the conference. Final papers are limited to 20 pages, including diagrams, figures and appendixes, and must be in troff, ASCII, or LaTeX format. We will supply you with instructions. Papers should include a brief description of the site, where appropriate.

Conference proceedings, containing all refereed papers and materials from the invited talks, will be distributed to attendees and will also be available from USENIX following the conference.

WHERE TO SEND SUBMISSIONS

Please submit extended abstracts for the refereed paper track by two of the following methods. All submissions will be acknowledged.

- Email to: ***lisa9papers@usenix.org***; Fax to: 510 548 5738; Mail to:
  LISA '95 Conference, USENIX Association, 2560 Ninth Street, Suite 215,
  Berkeley CA USA 94710

To discuss potential submissions, and for inquiries regarding the content of the conference program, contact the program co-chairs at ***lisa9chair@usenix.org*** or at:

- Tina M. Darmohray, 1630 Buena Vista Ave., Livermore CA USA 94550.
  510 443 4425; Fax: 510 962 0842; Email: ***tmd@greatcircle.com***
- Paul Evans, Synopsys, Inc., 700 East Middlefield Road, Mountain View
  CA USA 94043. 415 694 1855; Fax: 415 965 8637;
  Email: ***ple@usenix.org***

INVITED TALKS TRACK

If you have a topic of general interest to system administrators, but that is not suited for a traditional technical paper submission, please submit a proposal for a second track presentation to the invited talk (IT) coordinators at ***itlisa@usenix.org*** or to:

- Laura de Leon, Hewlett-Packard. 415 857 5605; Fax: 415 857 5686;
  Email: ***deleon@hpl.hp.com***
- Peg Schafer, Harvard. 617 495 4927; Fax: 617 496 5508;
  Email: ***peg@harvard.edu***

LISA'95

SEPTEMBER 18–22, 1995

PROGRAM COMMITTEE

- Program Co-chair: Tina Darmohray, *Consultant*
- Program Co-chair: Paul Evans, *Synopsys, Inc.*

Paul Anderson, *University of Edinburgh*
Kim Carney, *Massachusetts Institute of Technology*
Rob Kolstad, *Berkeley Software Design, Inc.*
Bryan McDonald, *SRI International*
Marcus Ranum, *Trusted Information Systems, Inc.*
John Schimmel, *Silicon Graphics, Inc.*

VENDOR DISPLAY

Wed. & Thurs., September 20–21, 1995

Well-informed vendor representatives will demonstrate products and services at the informal display. If your company would like to participate, please contact:

- Peter Mui
  Phone: 510 528 8649
  Fax: 510 548 5738
  Email: ***display@usenix.org***

FOR CONFERENCE REGISTRATION INFORMATION

All details of the technical and tutorial programs, registration fees and forms, and hotel information will be available in July, 1995. If you wish to receive the registration materials, please contact USENIX:

- USENIX Conference Office
  22672 Lambert Street, Suite 613
  Lake Forest, CA 92630 USA

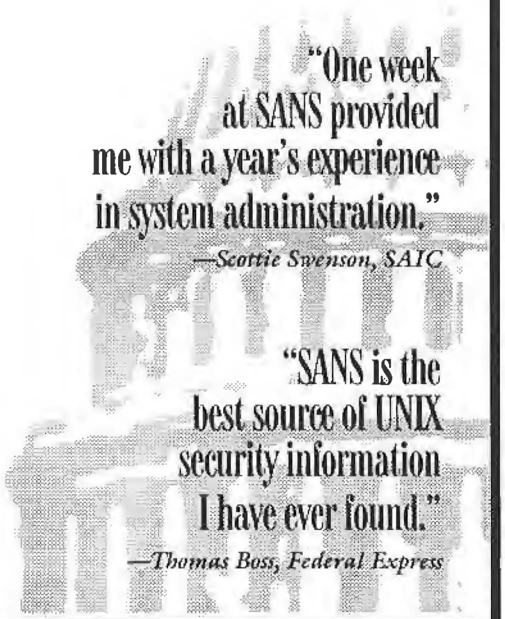
Phone: 714 588 8649
Fax: 714 588 9706
Email: ***conference@usenix.org***

For more information about USENIX and its events, access the USENIX Resource Center on the World Wide Web. The URL is ***http://www.usenix.org***. Or send email to our mailserver at: ***info@usenix.org***. Your message should contain the line: ***send catalog***. A catalog will be returned to you.

CO-SPONSORED BY:

USENIX® AND SAGE THE SYSTEM ADMINISTRATORS GUILD

The Advanced Computing
Systems Professional and
Technical Association



Computer Publications from John Wiley & Sons, Inc.

USENIX members receive
a 15% discount

The Internet Navigator, 2ed
Paul Gilster
1-05260-4 $24.95 **member price: $21.20**
# of copies: ________

Advanced Topics in UNIX
Ronald Leach
1-03663-3 $24.95 **member price: $21.20**
# of copies: ________

Introduction to Client Server Systems
Paul Renaud
1-57774-X $34.95 **member price: $29.70**
# of copies: ________

Portable UNIX
Douglas Topham
1-57926-2 $14.95 **member price: $12.71**
# of copies: ________

UNIX, Self-Teaching Guide
George Leach
1-57924-6 $19.95 **member price: $16.95**
# of copies: ________

Object Oriented Programming with Turbo C++
Keith Weiskamp
1-52466-2 $24.95 **member price: $21.20**
# of copies: ________

Obfuscated C and Other Mysteries
Don Libes
1-57805-3 $39.95 **member price: $33.96**
# of copies: ________

Finding It On the Internet
Paul Gilster
1-03857-1 $19.95 **member price $16.95**
# of copies: ________

Internationalization: Developing Software for Global Markets 1-07661-9 ***(pub. date: 1/95)***
Tuoc Luong $29.95 **member price: $25.45**
# of copies: ________

Adventures in UNIX Network Applications Programming
Bill Rieken
1-52858-7 $39.95 **member price: $33.96**
# of copies: ________

UNIX Shell Programming, 3e
Lowell Jay Arthur
1-59941-7 $29.95 **member price: $25.45**
# of copies: ________

The UNIX Command Reference Guide
Kaare Christian
1-85580-4 $32.95 **member price: $28.01**
# of copies: ________

Berkeley UNIX: A Simple & Comprehensive Guide
James Wilson
1-61582-X $40.95 **member price: $34.80**
# of copies: ________

□ Payment enclosed, plus sales tax
□ VISA □ Mastercard
□ American Express
Card No. ________________

Name ________________
Firm ________________
Address ________________
City/State/Zip ________________
Signature ________________
(order invalid unless signed

Please send all orders to:

John Wiley & Sons, Inc.
Attn: Karen Cooper, Special Sales
605 Third Avenue
New York, NY 10158
Phone: (212) 850-6789
Fax: (212) 850-6142

D582

Prentice Hall PTR is pleased to recommend the following titles to USENIX members...

USENIX members receive a 15% discount on orders

____**UNIX System Administration Handbook, Second Edition,** Evi Nemeth/Garth Snyder, 0-13-151051-7
(15105-0) *List:* $48.00 *Members:* $40.80

____**Object-Oriented Modeling and Design,** James Rumbaugh, 0-13-629841-9
(62984-0) *List:* $54.00 *Members:* $45.90

____**Zen and the Art of the Internet, Third Edition,** Brendan Kehoe, 0-13-121492-6
(12149-1) *List:* $23.95 *Members:* $20.36

____**The Magic Garden Explained,** Bernard Goodheart/James Cox, 0-13-098138-9
(09813-7) *List:* $38.00 *Members:* $32.30

____**Internetworking with TCP/IP, Vol. II Design, Implementation, and Internals,** Douglas E. Comer/David L. Stevens, 0-13-472242-6
(47224-1) *List:* $61.33 *Members:* $52.13

____**SCO Performance Tuning Handbook,** Gina Miscovich/David Simons, 0-13-102690-9
(10269-9) *List:* $42.00 *Members:* $35.70

____**Object-Oriented Programming,** Peter Coad/Jill Nicola, 0-13-032616-X
(03261-5) *List:* $48.00 *Members:* $40.80

____**Internetworking with TCP/IP, Vol. III Client Server Programming and Applications for the BSD Socket Version,** Douglas E. Comer and David L. Stevens, 0-13-474222-2
(47422-1) *List:* $53.00 *Members:* $45.05

____**Internetworking with TCP/IP, Vol. III Client Server Programming and Applications for the AT&T TLI Version,** Douglas E. Comer and David L. Stevens, 0-13-474230-3
(47423-9) *List:* $53.00 *Members:* $45.05

____**The Internet Message: Closing the Book with Electronic Mail,** Marshall T. Rose, 0-13-092941-7
(09294-0) *List:* $50.00 *Members:* $42.50

____**The Standard C Library,** PJ Plauger, 0-13-131509-9
(13150-8) *List:* $37.80 *Members:* $32.13

____**All About Administering the NIS+, Second Edition** Rick Ramsey, 0-13-309576-2
(30957-5) *List:* $42.00 *Members:* $35.70

____**The Simple Book: An Introduction to Internet Management** Marshall T. Rose, 0-13-177254-6
(17725-3) *List:* $55.00 *Members:* $46.75

____**Networking Operations on UNIX SVR4,** Mike Padavano, 0-13-613555-2
(61355-4) *List:* $50.00 *Members:* $42.50

____**Solaris Porting Guide,** Sunsoft ISV Engineering 0-13-030396-8
(03039-5) *List:* $52.00 *Members:* $44.20

____**Multiprocessor System Architectures,** Ben Catanzaro 0-13-089137-1
(08913-6) *List:* $44.00 *Members:* $37.40

____**The HP-UX System Administrator's "How To" Book** Marty Poniatowski, 0-13-099821-4
(09982-0) *List:* $32.00 *Members:* $27.20

____**UNIX System V Performance Management,** Edited by Phyllis Eve Bregman and Sally A. Browning 0-13-016429-1
(01642-8) *List:* $29.95 *Members:* $25.45

____**SCO® UNIX® Operating System System Administrator's Guide,** Santa Cruz Operation, 0-13-012568-7
(01256-7) *List:* $39.00 *Members:* $33.15

HERE'S HOW TO ORDER:

CALL
800-880-6818

OR WRITE:

CompuBooks
Route 1, Box 271-D,
Cedar Creek, TX 78612

OR INTERNET:

70007.1333@CompuServe.com
(GO CBK on CompuServe)

WE SHIP ANYWHERE!

FOR MORE INFORMATION, OR QUANTITY ORDERS, PLEASE CALL 201-592-2657

A UNIQUE OFFER ON THE BEST IN UNIX FOR USENIX MEMBERS

□ **THE INTERNET GUIDE FOR NEW USERS**
D. Dern
hardcover, 016510-6, $40.00,
MEMBER PRICE $32.00
paperback, 016511-4, $27.95,
MEMBER PRICE $22.36

□ **INTERNET FOR EVERYONE**
R. Wiggins
hardcover, 067018-8, $29.95,
MEMBER PRICE $23.96
paperback, 067019-6, $45.00,
MEMBER PRICE $36.00

□ **THE ESSENTIAL INTERNET INFORMATION GUIDE**
J. Manger
707905-1, paperback, $27.95,
MEMBER PRICE $22.36

20% DISCOUNT FROM McGRAW-HILL

□ **THE INFORMATION BROKERS HANDBOOK, Second Edition**
S. Rugge
911878-x, paperback, $34.95,
MEMBER PRICE $27.96
Available December l994

□ **SAA AND UNIX: IBM's Open System Strategy**
M. Killen
034607-0, $40.00,
MEMBER PRICE $32.00

□ **A STUDENT'S GUIDE TO UNIX**
H. Hahn
025511-3, paperback, $28.00,
MEMBER PRICE $22.40

□ **UNIX DEVELOPER'S TOOL KIT**
K. Leininger
911836-4, $65.00,
MEMBER PRICE $52.00

□ **UNIX SECURITY: A Practical Tutorial**
N. Arnold
002560-6, $24.95,
MEMBER PRICE $19.96

□ **THE UNIX AUDIT: Using UNIX to Audit UNIX**
M. Grottola
025127-4, $32.95,
MEMBER PRICE $26.36

□ **UNIX: A Database Approach**
S. Das
015745-6, $29.95,
MEMBER PRICE $23.96
Available November l994

I am a member of USENIX Association. Please send me the books I have indicated at the member special rate. I have added $3.00 postage and handling for the first book ordered, $1.00 for each additional book, plus my local sales tax.

Check or money order is enclosed—payable to McGraw-Hill, Inc.
Charge my ❑ Visa ❑ Mastercard
❑ Discover ❑ Amex

Account #________________________

Expiration Date________________________

Signature________________________

USENIX Membership # ________________

Bill & Ship To:

Name________________________

Street________________________

City, State, Zip________________________

Daytime Phone #________________________

03US002

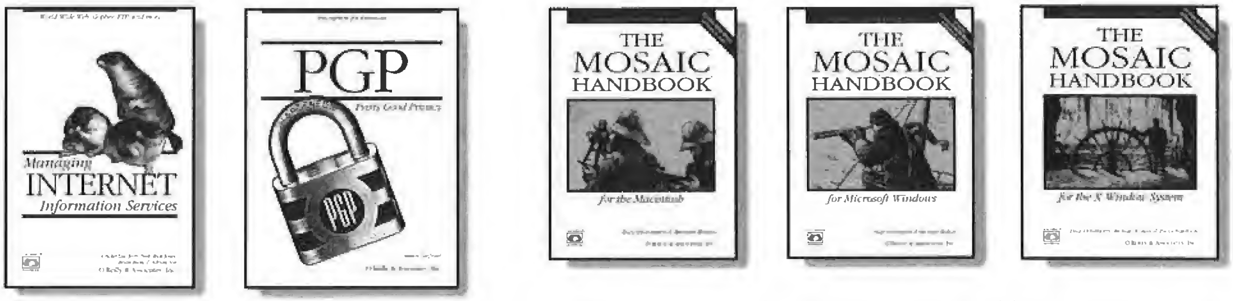
Send or Fax Orders to:
McGraw-Hill, Inc.
Attn: Rosa Perez
11 West 19th Street—4th Floor
New York, New York 10011
Fax: 212-337-4092

If I am not completely satisfied, I will return the book(s) within 15 days for a full refund or credit. Satisfaction unconditionally guaranteed. Prices subject to change without notics. We can only accept orders within the continental USA.

Commemorating 25 years of UNIX

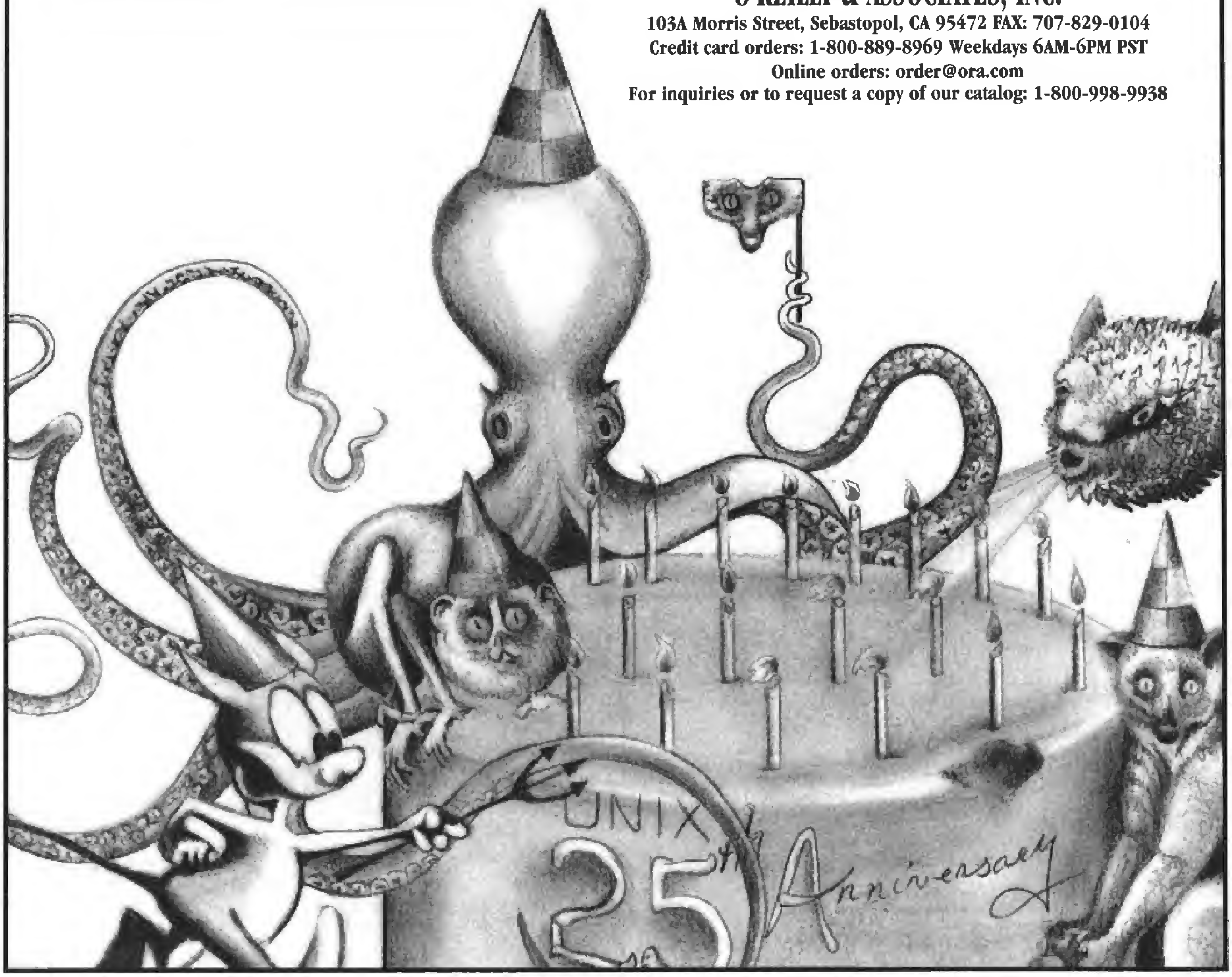
This year UNIX marks its 25th anniversary. We think its something to celebrate. After all, we owe our livelihoods to UNIX. Over the years it's given us plenty to explore, learn, teach, and talk about. It's enabled us to do good work—work that's filled with adventure, fun, and creative challenge. We couldn't ask for much more than that. Except, of course, for a little cake and ice cream.

You might like to celebrate with these new titles from O'Reilly: *Managing Internet Information Services*, *The Mosaic Handbook* (with software for three platforms, including Microsoft Windows, X, and the Macintosh), *X User's Tools*, *Motif Tools*, *PGP: Pretty Good Privacy*, and *Multi-Platform Code Management*. Call or write to us at the numbers below for details. Mention code ALOG.



O'REILLY & ASSOCIATES, INC.

103A Morris Street, Sebastopol, CA 95472 FAX: 707-829-0104
Credit card orders: 1-800-889-8969 Weekdays 6AM-6PM PST
Online orders: order@ora.com
For inquiries or to request a copy of our catalog: 1-800-998-9938



TAME YOUR MASSIVE OPEN SYSTEMS ENVIRONMENT: Attend The MOSES Conference

Sponsored by UniForum Association and The MOSES Group

The MOSES GROUP:

An alliance of xperienced ractitioners share heir knowledge

Oracle Corporation

Sequent Computer Systems, Inc.

Fujitsu Computer Products of America

Millipore Corporation

U S WEST NewVector Group, Inc.

British Telecom

Burlington Coat Factory

Quantum

***lus* a panel of 30 xperts from eading companies hare their olutions.**

Moderated by
Information Week
(incorporating
Open Systems Today)

Co-Sponsored by
Sequent Computer Systems, Inc.

SEQUENT
Our Business Is Your Success

LOCATIONS & DATES:

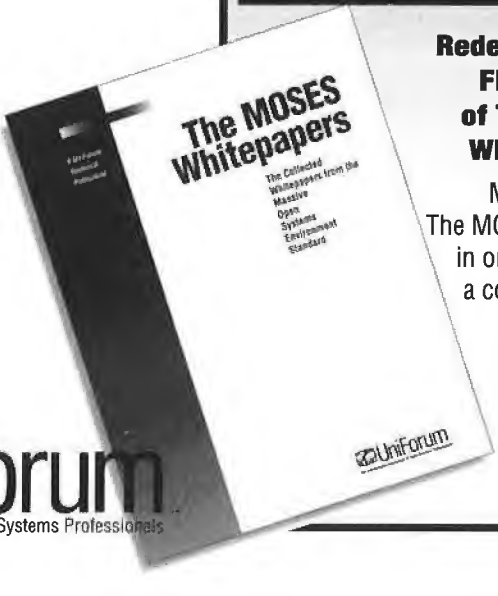
UniForum Annual Conference
Dallas Convention Center
March 12-13, 1995

Boston World Trade Center
Boston, MA
May 2-3, 1995

If you're serious about moving to, managing, and maintaining a large, relational database in an open systems environment, the Massive Open Systems Environment Standard Conference is for you!

Special Exhibition Showcase highlighting systems management solutions at Boston event.

CONFERENCE ATTENDEES


The MOSES Whitepapers
The Collected Whitepapers from the Massive Open Systems Environment Standard
UniForum

Redeemable for a FREE COPY of The MOSES Whitepapers

Must attend
The MOSES Conference
in order to receive
a complimentary
copy.

$25.00 VALUE

REGISTER TODAY!

CALL 1-800-255-5620

EXTENSION 34

UniForum
2901 Tasman Drive
Suite 205
Santa Clara, CA 95054
800-255-5620 • 408-986-8840
408-986-1645 fax
URL http://www.uniforum.org

UniForum
e International Association of Open Systems Professionals

What is Open Systems and where can you find the information you need to keep up in today's competitive open systems environment?

Find out by joining UniForum: The International Association of Open Systems Professionals.

The Advantages of Membership:

Your annual membership is a resource treasure chest. Among the many valuable benefits you'll receive are:

- *UniForum Monthly*: The Magazine for Open Systems Professionals
- *UniNews* - the authoritative voice of UniForum - now on-line
- *The Open Systems Products Directory* - comprehensive, accurate, indispensable
- Reduced fees at all UniForum conferences and seminars
- Technical and Standards publications: timely, reliable and useful
- Discounts on many outstanding industry publications
- Discounts on leading commercial hardware and software products
- Discounts on courses offered by leading training organizations
- Discounts on exclusive new shareware selections
- Services such as discounts on car rentals, travel and mail list rentals
- Eligibility to serve on UniForum committees and the Board of Directors

UNIFORUM MEMBERSHIP APPLICATION FORM *(please print)*

Name ______________________________ MR/MS

Title ______________________________

Company ______________________________

Address ______________________________

Mail Stop ______________________________

City ______________ State ______________ Zip ______________

Country ______________ Telephone (______) ______________

E-mail (where we will send UniNews) ______________ Fax (______) ______________

General Membership** $125

Includes *UniForum Monthly* magazine, *UniNews* electronic newsletter, *Open Systems Products Directory,* technical publications, recruitment advertising service, and discounts on: UniForum conference registration and shareware CD-ROMs, purchases of software and hardware, educational seminars and special classes, additional UniForum publications and other industry publications, computer books, CD-ROMs and training services.

** Overseas postage:
air $100 or surface $60 (no additional postage necessary for Canada, Mexico or Puerto Rico) $__________

TOTAL AMOUNT ENCLOSED $__________

Method of Payment:

Select one: □ Check payable to UniForum *□ Money Order In US Dollars
□ MasterCard □ Visa □ American Express

Credit card number ______________________ Expiration date ______________

Signature ______________________________

*Payments must be included in U.S. dollars. All checks must be drawn on a U.S. bank. Credit card orders can be accepted by telephone.
**Overseas surface delivery takes up to six weeks.

Send application and payment to: UniForum, 2901 Tasman Drive, Suite 205, Santa Clara, CA 95054-1100
Tel: (408) 986-8840 (800) 255-5620 Fax: (408) 986-1645

Prices subject to change without notice. Contact UniForum for discounts and shipping and handling charges on bulk orders.
UniForum is a registered trademark of UniForum. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

UniForum
The International Association of Open Systems Professionals

There's Never Been a Better Time for an Outstanding Value

As the unstoppable move to open systems gathers even more momentum you need a strong and independent association that can help you achieve your professional goals.
An association that can magnify your voice, which can educate, inform and inspire.
You need and deserve an association that provides value for your money. In short you need UniForum.
With a value this good, the real question might be, "Can you afford to be without it?"
Join today!

SUBSCRIPTION ORDER FORM

USENIX

PROCEEDINGS

Please enter my one-year subscription* to the 1995 USENIX Conferences and Symposia Proceedings, which includes:

New Orleans Technical Conference - January 1995

Mobile and Location-Independent Computing Symposium - April 1995

Fifth USENIX UNIX Security Symposium - June 1995

Conference on Object-Oriented Technologies (COOTS) - June 1995

Tcl/Tk Workshop - July 1995

Workshop on Electronic Commerce - July 1995

Systems Administration (LISA '95) Conference - September 1995

*NOTE:
Corporate, Educational and Supporting Members of USENIX automatically receive a subscription to all proceedings published during their term of membership, which may overlap with this offer.

COST: USENIX members: $120.00
Non-members: $160.00
Outside U.S.A. and Canada?
Add $30.00 for surface postage.

Subscription Cost	______ **
Calif. Sales Tax	______
Overseas Postage	______
TOTAL ENCLOSED	______

*** If you are paying member price, please include member's name and/or membership number*______

PAYMENT OPTIONS

□ Check enclosed payable to USENIX Association. □ Purchase order enclosed.
□ Please charge my: □ Visa □ MasterCard

Account # ______ Exp. Date ______

Signature ______

Outside the U.S.A.? Please make your payment in U.S. currency by one of the following:
- * Check - issued by a local branch of a U.S. Bank
- * Charge (Visa, MasterCard, or foreign equivalent)
- * International postal money order

2/95

Name ______

Address ______

City ______ State/Country ______ Zip/Postal Code ______

Please mail this order form to: USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710
510/528-8649
FAX 510/548-5738
office@usenix.org

This offer is good through December 31, 1995

The proceedings are shipped the week following the event.

LOCAL USER GROUPS

The Association will support local user groups by doing a mailing to assist in the formation of a new group and publishing information on local groups in *;login:*. At least one member of the group must be a current member of the Association. Send additions and corrections to: <*login@usenix.org*>.

California

Fresno:

The Central California UNIX Users Group consists of a uucp-based electronic mailing list to which members may post questions or information. For connection information:

- Educational and governmental institutions:
  Brent Auernheimer
  (209) 278-2573,
  <*brent@CSUFresno.edu*> or
  <*csufres!brent*>
- Commercial institutions or individuals:
- Gordon Crumal
  (209) 251-2648
  <*csufres!gordon*>

Orange County

Meets the 2nd Monday of each month

- UNIX Users Association of Southern California
  Dave Close
  (714) 434-7359
  <*dhclose@alumni.caltech.edu*>
  New Horizons Computer Learning Center
  1231 E. Dyer Rd., Suite 140
  Santa Ana, CA 92705
  (714) 438-9440

Colorado

Boulder

Meets monthly at different sites. For meeting schedule, send email to <*fruug-info@fruug.org*>.

- Front Range UNIX Users Group
  Lone Eagle Systems Inc.
  636 Arapahoe #10
  Boulder, CO 80302
  Steve Gaede
  (303) 444-9114
  <*gaede@fruug.org*>
  <*http://www.fruug.org/~fruug*>

Washington, D.C.

Meets 1st Tuesday of each month.

- Washington Area UNIX Users Group
  9811 Mallard Drive
  Laurel, MD 20708
  Alan Fedder
  (301) 953-3626

Florida

Coral Springs:

- S. Shaw McQuinn
  (305) 344-8686
  8557 W. Sample Road
  Coral Springs, FL 33065

Melbourne:

Meets the 3rd Monday of every month.

- Space Coast UNIX User's Group
  Steve Lindsey
  (407) 242-4766
  <*lindsey@vnet.ibm.com*>

Orlando:

Meets the 3rd Thursday of each month.

- Central Florida UNIX Users Group Mikel Manitius
  (407) 444-8448
  <*mikel@aaa.com*>

Western:

Meets 1st Thursday of each month.

- Florida West Coast UNIX Users Group
  Richard Martino (813) 536-1776
  Tony Becker (813) 799-1836
  <*mcrsys!tony*>
  Ed Gallizzi, Ph.D. (813) 864-8272
  <*e.gallizzi@compmail.com*>
  Jay Ts (813) 979-9169
  <*uunet!pdn!tscs!metran!jan*>
  Dave Lewis (407)242-4372
  <*dhl@ccd.harris.com*>

Georgia

Atlanta:

Meets on the 1st Monday of each month in White Hall, Emory University.

- Atlanta UNIX Users Group
  P.O. Box 12241
  Atlanta, GA 30355-2241
  Mark Landry (404) 365-8108

Kansas or Missouri

Meets on 2nd Tuesday of each month.

- Kansas City UNIX Users Group (KCUUG)
  P.O. Box 412622
  Kansas City, MO 64141
  (816) 891-1093
  <*richj@northcs.cps.com*>

Michigan

Detroit/Ann Arbor

Meets on the 2nd Thursday of each month in Ann Arbor.

- Southeastern Michigan Sun Local Users Group and Nameless UNIX Users Group
  Steve Simmons' office: (313)769-4086
  home: (313) 426-8981
  <*scs@lokkur.dexter.mi.us*>

Minnesota

Minneapolis/St. Paul:

Meets the 1st Wednesday of each month.

- UNIX Users of Minnesota
  17130 Jordan Court
  Lakeville, MN 55044
  Robert A. Monio
  (612) 220-2427
  <*pnessutt@dmshq.mn.org*>

Missouri

St. Louis:

- St. Louis UNIX Users Group
  P..O. Box 2182 St. Louis, MO 63158
  Terry Linhardt
  (314) 772-4762
  <*uunet!jgaltstl!terry*>

Nebraska

Omaha:

Meets monthly.

- /usr/group/nebraska
  P.O. Box 31012
  Omaha, NE 68132
  Phillip Allendorfer
  (402) 423-1400

New England

Northern:

Meets monthly at different sites.

- Peter Schmitt 603) 646-2085
  Kiewit Computation Center
  Dartmouth College Hanover, NH 03755
  <*peter.schmitt@dartmouth.edu*>

New Jersey

Princeton:

Meets monthly.

- Princeton UNIX Users Group
  Mercer County Community College
  1200 Old Trenton Road
  Trenton, NJ 08690
  Peter J. Holsberg (609) 586-4800
  <mccc!pjh>

New Mexico

Albuquerque:

ASIGUNIX meets every 3rd Wednesday of each month.

- Phil Hortz (505) 275-0466.

New York

New York City:

Meets every other month in Manhattan.

- Unigroup of New York City
  G.P.O. Box 1931
  New York, NY 10116
  <unigroup@murphy.com>
  Bob Young (212) 490-8470

Oklahoma

Tulsa:

Meets 2nd Wednesday of each month.

- Yulsa UNIX Users Group, $USR
  Stan Mason (918) 560-5329
  <tulsix!smason@drd.com>
  Mark Lawrence (918) 743-3013
  <mark@drd.com>

Texas

Austin:

Meets 3rd Thursday of each month.

- Capital Area Central Texas UNIX Society (CACTUS)
  P.O. Box 9786
  Austin, TX 78766-9786
  Tom Painter (512) 258-7321
  <president@cactus.org>

Dallas/Fort Worth:

Meets the 1st Thursday of each month.

- Dallas/Fort Worth UNIX Users Group
  P.O. Box 867405
  Plano, TX 75086
  Evan Brown (214) 519-3577
  <evbrown@dsccc.com>

Houston:

Meets 3rd Tuesday of each month.

- Houston UNIX Users Group (Hounix) answering machine
  (713) 684-6590
  Bob Marcum, President
  (713) 626-4100
  Chuck Bentley, Vice-president
  (713) 789-8928
  <chuckb@hounix.uucp>

Washington

Seattle:

Meets monthly.

- Seattle UNIX Group Membership Info.
  Bill Campbell (206) 947-5591
  6641 East Mercer
  Mercer Island, WA 98040-0820
  <bill@celestial.com>

Canada

Manitoba:

Meets 2nd Tuesday of each month.

- Manitoba UNIX User Group (MUUG) P.O. Box 130
  St. Boniface Winnipeg,
  MB R2H 3B4
  Bary Finch, President
  (204) 934-2723
  <info@muug.mb.ca>

Ottawa:

- The Ottawa Carleton UNIX Users Group
  D.J. Blackwood
  (613) 957-9305
  <dave@revcan.rct.ca>

Toronto:

- 143 Baronwood Court
  Brampton, Ontario
  Canada L6V 3H8
  Evan Leibovitch
  (416) 452-0504
  <evan@telly.on.ca>

Quebec:

Meetis first Wednesday every 3rd month.

- Administrateurs de Systeme UNIX du Quebec (ASUQ)
  Universite de Montreal, Dept. IRO
  C.P. 6128, Succ. Centre-Ville
  Montreal, Quebec, Canada, H3C 3J7
  Tel: (514) 343-7480
  Fax: (514) 343-5834

System Administration Groups

Back Bay LISA (BBLISA)

New England forum covering all aspects of system and network administration, for large and small installations. Meets monthly, at MIT in Cambridge, MA.

For information, contact:

- J. R. Oldroyd (617) 227-5635
  <jr@opal.com>
- Mailing list subscription:
  <bblisa-request@bblisa.org>
- Mailing list postings:
  <bblisa@bblisa.org>
- For current calendar of events:
  finger *<bblisa@finger.bblisa.org>*

Bay LISA (California)

Meets 3rd Thursday of each month in Mountain View, CA For more information, please contact: *<baylisa-info@baylisa.org>*

$GROUPNAME (New Jersey)

$GROUPNAME is an organization in New Jersey formed to facilitate information exchange pertaining to the field of UNIX system administration. For more information, send email to:
Majordomo@Warren. MENTORG.COM or
Tom Limoncelli
<tom_limoncelli@warren.mentorg.com>

New York Systems Administrators (NYSA)

Meets 2nd Monday of each month.

- *<nysa-request@esm.com>*
  (914) 472-3635 or 472-3635

North Carolina System Administrators Group

The North Carolina System Administrators Group meets on the 2nd Monday each month around the Research Triangle Park area.

- Amy Kreiling (919) 962-1843
  <kreiling@cs.unc.edu>
- William E. Howell (919) 962-1717
  <howell@cs.unc.edu>

CALENDAR OF EVENTS

This is a combined calendar of conferences, symposia, and standards meetings. If you have a event that you wish to publicize, please contact <*login@usenix.org*>.

* = events sponsored by the USENIX Association.

1995

April

3 -7	IETF, Danvers, MA
10-11*	2nd Mobile and Location--Independent Computing, Ann Arbor, MI
10-14	IEEE 1003
10-14	3rd International World Wide Web Conference, Darmstadt, Germany
24-29 *	SANS IV, Washington, DC

May

4 - 5	ACM 5th Workshop on Hot Topics in Operating Systems Orcas Island, WA
5 - 11	DECUS, Washington, DC
17-20	UniForum NZ, Masterton, New Zealand
22-25	SunWorld Expo, San Francisco, CA
29-J2	NetWorld + Interop 95, Frankfurt, Germany

June

5 - 7*	UNIX Security, Salt Lake City, UT
18-23	ACM SIGPLAN Conference, La Jolla, CA
26-29*	COOTS, Monterey, CA

July

6-11*	Tcl/Tk Workshop, Toronto, Canada
10-14	IEEE 1003
17-21	IETF, Stockholm, Sweden
17-21	NetWorld + Interop 95, Tokyo, Japan

August

6-11	ACM SIGGRAPH, Los Angeles, CA
14-18	Interex 95, Toronto, Canada
30-S1	ACM SIGCOMM '95, Cambridge, MA

September

12-14	GUUG Annual Conference, Weisbaden, Germnany
18-21	AUUG, Sydney, Australisa
18-22*	LISA '95, Monterey, CA
19-21	UNIX Expo, New York City
25-29	NetWorld + Interop 95, Atlanta, GA

October

9-13	IEEE 1003
12-15	SIGSOFT '95, Washington, DC
15-19	OOPSLA '95, Austin, TX
25-28	IEEE Parallel & Distributed Processing Symposium, San Antonio, TX

November

2 - 8	DECUS, San Francisco, CA
5 - 9	ACM Multimedia '95, San Francisco, CA
6 -10	NetWorld + Interop 95, Paris

December

2 - 7	DECUS, San Francisco, CA
3 - 6	SOSP, Colorado
4 - 8	IETF, Dallas, TX
4 - 8	ACM/IEEE CS Supercomputing '95, San Diego, CA
11-14	4th WorldWideWeb Conference, Boston, MA

1996

January

22-26*	USENIX, San Diego, CA

February

14-16	UniForum, San Francisco, CA

May

18-24	DECUS, Orlando, FL

August

4 - 8	Interex '96, San Diego, CA

September

AUUG, Melbourne, Australia

October

8-10	UNIX Expo, New York City

November

9-14	DECUS, Anaheim, CA

December

JUS UNIX Fair, Tokyo, Japan

ACM:
Association for Computing Machinery

AFUU:
Association of French UNIX Users

ASPLOS:
Architectural Support for Programming Languages and Operating Systems

AUUG:
Australian UNIX Systems Users Group

COOTS:
Conference on Object-Oriented Technologies

DECUS:
Digital Equipment Computer Users Society

EurOpen:
European Forum for Open Systems

GURU:
Romanian UNIX User Group

GUUG:
German UNIX Users Group

IEEE: Institute of Electrical and Electronics Engineers

IETF:
Internet Engineering Task Force

Interex:
International Association of Hewlett-Packard Computer Users

JUS:
Japan UNIX Society

LISA:
USENIX/SAGE Systems Administration Conference

OOPSLA:
Object-oriented Programming Systems, Languages and Applications

OSDI:
Symposium on Operating Systems Design & Implementation

SANS:
System Administration, Networking & Security

SIGSOFT:
ACM Special Interest Group on Software Engineering

SOSP:
Symposium on Operating Systems Principles

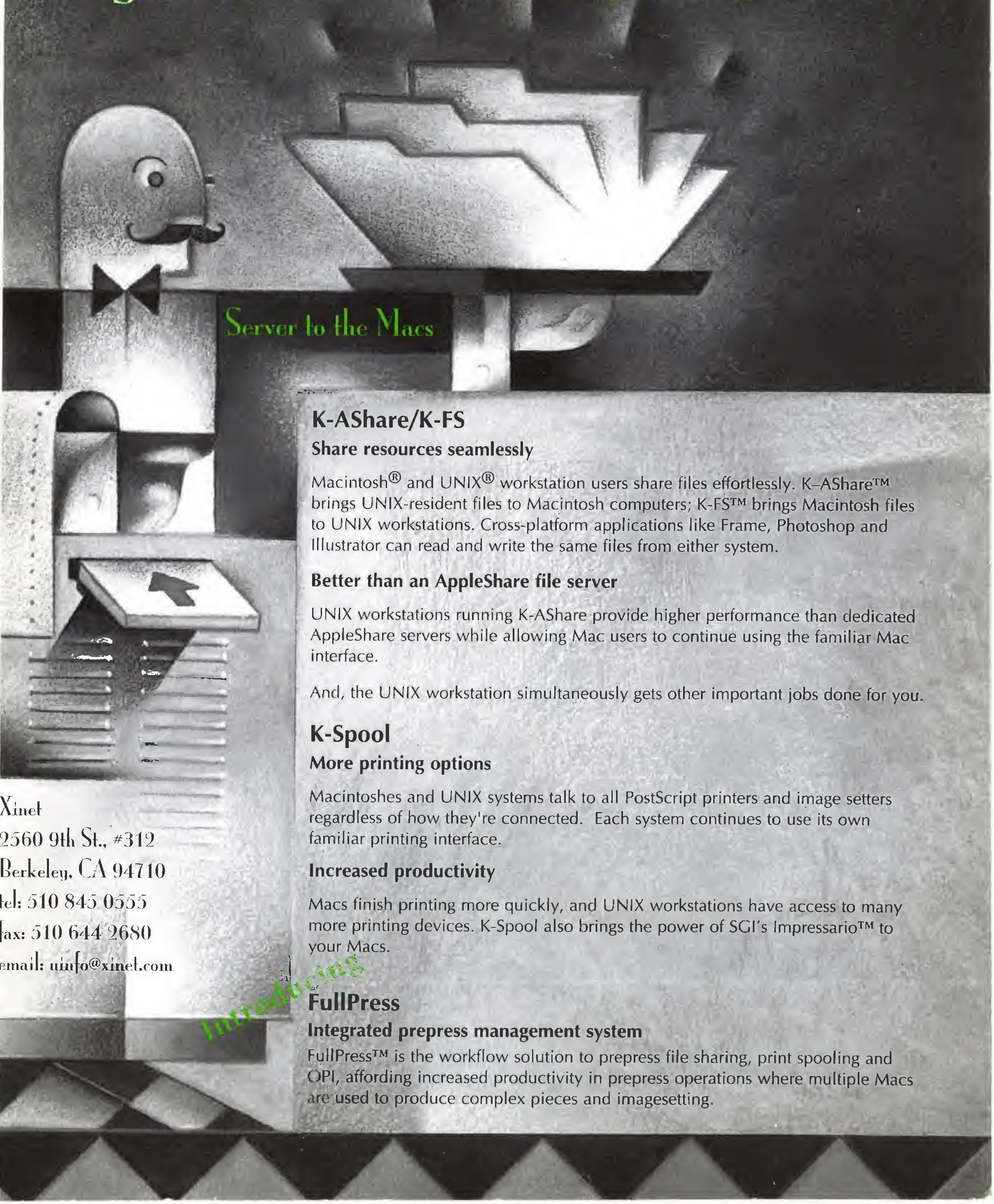
SUG:
Sun User Group

SUUG:
Society of Russia UNIX Users Group

UKUUG:
United Kingdom UNIX Systems Users Group

UniForum:
International Association of UNIX and Open Systems Professionals

Integrate Macs with Sun, SGI, & HP



K-AShare/K-FS

Share resources seamlessly

Macintosh® and UNIX® workstation users share files effortlessly. K–AShare™ brings UNIX-resident files to Macintosh computers; K-FS™ brings Macintosh files to UNIX workstations. Cross-platform applications like Frame, Photoshop and Illustrator can read and write the same files from either system.

Better than an AppleShare file server

UNIX workstations running K-AShare provide higher performance than dedicated AppleShare servers while allowing Mac users to continue using the familiar Mac interface.

And, the UNIX workstation simultaneously gets other important jobs done for you.

K-Spool

More printing options

Macintoshes and UNIX systems talk to all PostScript printers and image setters regardless of how they're connected. Each system continues to use its own familiar printing interface.

Increased productivity

Macs finish printing more quickly, and UNIX workstations have access to many more printing devices. K-Spool also brings the power of SGI's Impressario™ to your Macs.

FullPress

Integrated prepress management system

FullPress™ is the workflow solution to prepress file sharing, print spooling and OPI, affording increased productivity in prepress operations where multiple Macs are used to produce complex pieces and imagesetting.

Xinet
2560 9th St., #312
Berkeley, CA 94710
tel: 510 845 0555
fax: 510 644 2680
email: uinfo@xinet.com

;login:

USENIX Association
2560 Ninth Street
Suite 215
Berkeley, CA 94710

POSTMASTER:
SEND ADDRESS CHANGES
TO *;login:*
USENIX ASSOCIATION
2560 NINTH STREET
SUITE 215
BERKELEY, CA 94710

Wanted:

Papers to publish

and authors

to present

(turn to pages 56-62)